DRAGON USER

The independent Dragon magazine

November 1987

Contents

APPEAL+APPEAL+APPEAL+APPEAL
In the struggle to get SEPTEMBER to its rightful owners, some people were sent duplicate copies. We now have a shortage in the office — so if anyone has a spare SEPTEMBER they can bear to part with, please invest 13p and send it back to us at your leisure.

Editorial

WELL, they didn't lynch me at Rochdale. But if they had I would not have been surprised. To all the readers who phoned or wrote to us, particularly to those who were told first one date and then another, thank you for bearing with us. Considering that she has been answering the phone to Dragon users almost back to back for the last three weeks, and faces another week of it while the issues crawl through the post, Carol is still speaking to me, so you must have been nice to her.

Thanks. It's been a very, very long time since we last missed our print date (if ever, indeed) and I hope it never ever happens again.

A small factor like someone leaving suddenly can end up affecting thousands of individuals who don't know what's happening, and wish it wasn't, not to mention those who think they know what's happening, and wish it wasn't.

At 1.14pm on Thursday 17th September the first reader PHONED US BACK TO SAY THAT HIS COPY HAS ARRIVED! Thank you a thousand times, Mr. Hartley. But I know some of you still hadn't received it by the 23rd. And the 17th was the publication day for the OCTOBER edition. No, I hope this never happens again.

Telephone number
(All departments)
437-4343

Editor
HELEN ARMSTRONG

Production Editors
DAVID PRIVETT/LINDA SLOMAN

Editorial Secretary
CAROL FRITH

Advertisement Manager
ATHENA PEERMAN

Administration
ANNE MARIE ALLEN

Marketing Manager
HELEN PERRY

Managing Editor
PETER WORLOCK

Publishing Director
JENNY IRELAND

Subscriptions
UK £14 for 12 issues
Overseas (surface) £20 for 12 issues
ISSN 0265-177. Telex: 296275
Dragon User, 12/13 Little Newport Street, London WC2H 7PP
US address: c/o Business Press International, 205 East 42nd St, New York, NY 10017
Published by Scot Press Ltd.

Typesetting and Production by Artext Limited, London NW1.
Printed by Headley Brothers Ltd. Ashford, Kent
Registered at the Post Office as a newspaper.
Dragon and its logo are trademarks of Eurohard Ltd.

How to submit articles

The quality of the material we can publish in *Dragon User* each month will, to a very great extent depend on the quality of the discoveries that you can make with your Dragon. The Dragon computer was launched on to the market with a powerful version of Basic, but with very poor documentation.

Articles which are submitted to *Dragon User* for publication should not be more than 3000 words long. All submissions should be typed. Please leave wide margins and a double space between each line. Programs should, whenever possible, be computer printed on plain white paper and be accompanied by a tape of the program.

We cannot guarantee to return every submitted article or program, so please keep a copy. If you want to have your program returned you must include a stamped addressed envelope.

Letters

This is your chance to air your views — send your tips, compliments and complaints to Letters Page, *Dragon User*, 12-13 Little Newport Street, London WC2H 7PP.

Hands off our OS-9

I HAVE read *Three Little Words* in the June 1987 issue of *Dragon User*, and I feel quite strongly about the negative comments made about *Stylograph*. It seems obvious that Mr. Merrick has not been able to master rudimentary OS-9.

I will not argue that Stylo can be enhanced by an 80 column screen display, and of course two drives are better than one. But Stylo is in the league of any PC word processor I have seen. For the comment on the double line feed, he should read David Rothery's article in the March 1987 issue, and check the DIP switches on his printer. One of these will control the line feed/carriage return mode. Reverse the switch and all will be well. You will have to reset the switch for listings in Dragon mode, but it is not necessary to ditch your current printer!

The volume of memory is governed by the user. To obtain 15K on a level one system, use a parameter value on the Stylo calling statement, eg Stylo letter £18K, which should give you 5 or 6 pages to work with. If your masterpiece is longer, you can unload what you have written to disc, and carry on writing, and the whole document will remain under the same filename. The limit to the size of your document is effectively the size of your formatted disc.

The Spell and Mail merge programs are most professional. The standard dictionary contains 40,000 words, and you can add as many as you like. Mail merge gives the ability to print multiple copies and make text substitutions while printing.

If you run a small business, OS-9 is most certainly suited. A bit of upgrade to your Dragon 64, like a second disc drive, proper monitor etc. is worth the investment. There are packages like *Cash and VAT* and *Stock Control*.

My wife who is a creative writer and not computerate was horrified when I told her the time had come to change word processors. With utilities provided by the OS-9 User Group I copies all her original works onto OS-9 formatted diskettes.

The first week was almost unbearable for us both, with frequent phone calls from home demanding how, and why and where, and that I return *Telewriter* to her. The second week was a dramatic change. My wife was creating new works with vigour, and checking the old one with the Spell program.

The moral is 'a little perseverance will reap great rewards'.

N.C.Grayson
56 Hawthorn Crescent
Cosham
Portsmouth
Hants PO6 2TP

Every month we will be shelling out a game or two, courtesy of Microdeal, to the reader/s who send the most interesting or entertaining letters. So send us your hints and your opinions, send us your hi-scores and suggestions. Send us your best Dragon stories. What d'you think we are, mind readers?!

EXTRA PUFF

Words of wisdom for the wise

I greatly enjoy struggling with Gordon Lee's posers. Not only that, I have found them a great help as an aid to programming.

When I bought my Dragon I had no knowledge of Basic programming, so I had to start from scratch. I did not find copying out programs from the manual very helpful. I found that the 'suck it and see' — sometimes known as the experimental — method helped me most. Things one finds out for oneself tend to stick in the mind.

Many of you *Dragon USer* readers have the advantage over me since I only manage to press one correct key in ten, and I can't spell. Because of this I have been officially certified the World's Worst Programmer! In spite of this, I have won the odd prize. So you one and all are in with a chance.

For those who would like to have a go, although it is worth having a good read of the Dragon manual, I recomment the 'try it and get it wrong' method.(*Otherwise known as the 'suck it and find its a Chinese salted prune' method — Ed.*) In the manual there is a list of all the Dragon commands.

Start with the PRINT command, because it helps if Dragon gives an output, then play with INPUT, and carry on from there. By experimenting one can find out what to do, even things not in the manuals.

A final point. With Gordon's posers, I always try to solve them by several quite different means. If I get two answers that are the same then they could be right. But, above all, trying different methods gives considerable insight into programming.

Good hunting!

Ted Newman
113 New Haw Road
Addlestone
Surrey
KT15 2DA

Looks like Ted wins again!

Stylo style

I HAVE read *Three Little Words* in the June issue and I think Mr. Merrick has not read the manuals carefully. All of the programs can be used successfully if understood. They have vast differences of operation, and since they have to work with a selection of printers, need time to be used to their best advantage.

Stylograph will not send codes over 128 as it was designed for a 7-bit bus which was the American standard at the time. *Minitext* is very slow becasue it is in Basic, but it is adaptable to different computers and printers. *Telewriter* is possibly the most powerful compact word processor I have seen, and though it does not have right justification or spelling correction as standard, it has features not found on main frame packages.

The 'embedded format codes' allow you to send any character in the range of 0 to 255 directly to the printer, allowing use of any character the printer can produce, on nearly any printer. Format or typeface can be changed during printing, allowing mixed faces or styles. The embedded codes should sometimes be followed with spaces to make them work properly.

Incidentally, as files generated by *Telewriter* are text files, they should not be named as backup files, but as updated files, ie FRED1/TEX, FRED'/TEX under DragonDOS. On DeltaDOS there are no bugs, so no problems arise.

If we can be of help to any readers we will provided they send as SAE.

A.C.Earnshaw
(High security systems)
23 Dellfield Crescent
Uxbridge
Middx. UB8 2ET

PS Can anyone help us? We have a Tandy version, but it has become corrupted and will no longer load.

Not so Mfree

Roger Merrick's article on word processors made some valid points, but I feel that he was less than fair to OS-9 and *Stylograph*.

OS-9 is complex and can be frustrating, but Stylo is automatic in its use of the OS-9 filing system, and you need know only a little to use it. A small business would surely find OS-9 with its utilities a godsend. Also, the idea of putting OS-9 in ROM would remove one

of the glories of OS-9 — the fact that the number of commands available is only limited by the size of your disc, and that extra utilities are readily available from the public domain through user groups.

I am puzzled by Mr. Merrick's use of MFREE to check the memory available. If you call this from Stylo, you are checking what is *left* to OS-9, not what is available to your text. You can call different amounts of memory for text, using STYLO #15K, for example, and the more you ask for, the less will show on MFREE! Also, OS-9 version 2.0 has a better screen driver, and is on buff, not green, which Mr. Merrick prefers. However, using the Compusense DragonPlus board gives a marvellous 80 column display and an almost instantaneous ramdisc on which your commands may be put. I would argue that even allowing for the hundred pounds or so for the add-on, you have a system which would compete favourable with much more expensive machines. Stylo can made *Telewriter* seem like a toy by comparison, useful as it is.

Telewriter on disc does right justify. Just press J in the format menu, and K to remove it. The problem with *Telewriter* is that it is all too easy to lose text — try printing out before saving, and with some printer settings you lose the lot. Also, if your disc is full, if you are lucky *Telewriter* will tell you. On the other hand, it might write all over your existing files, as mine did. (Yes, I DID have a backup, Helen!) And I would patiently line up tables of figures and select 80 column width for the printer, only to find that *Telewriter* had rearranged the lot beyond recognition. Stylo allows the line length to be set first, and the 'embedded format' commands are much more straightforward.

Incidentally, if you don't want OS-9's facilities, Flex is somewhat simpler to learn, and Flex's *SPEdit* is really Stylo under another name.

David Rothery
1, Heath Road
Glossop
Derbyshire
SK13 9AY

Once again, the OS-9 Users Group can be contact through Martin Vernon, 4 Roseberry Court, Llandudno, Gwynedd, LL30 1TF

Author gets it right

Thank you for publishing *Three Little Words*. A number of points have occurred to me since submitting this to you last year.

a) The problems of *Stylograph* are overcome by implementing the patches published in a recent issue of Dragon User.

b)I believe a three and a half inch disc uses a different standard and cannot be successfully connected to the Dragon disc cartridge.

c)Tandy CoCo OS-9 will not boot on the Dragon because the device drivers and the disc format are different.

d)I joined the OS-9 Users Group and have made more progress in OS-9 since then than in all the time before.

e)A 20 megabyte hard disc and interface would now cost around £350. When I bought the DragonDOS disc drive from Boots, it cost me ... £350.

Roger Merrick
30 Dean Road
Erdington
Birmingham
B23 6QF

Atoms are found again

In an old copy of *Dragon User* (June 1984) I found a program with the name 'In search of Atoms'. At first the description did not attract me, but I decided to try it. I have put in and tested most of the programs in DU but this is in my opinion the best. It is very good with the use of colour, and is easy to play. When playing it you must think quite a bit. If you have June 1984 and want a good game, try it.

This is not necessary, but I recommend you to change line 620 to:

```
620 R1=INT(SL*RND
(-TIMER))+1:
R2=INT(SL*RND(-TIMER))+1
```

Otherwise the same configuration comes up with every new start.

Martin Norrman
Edsbergsvagen 12
19151 Sollentuna
Sweden

News desk

If you have any new products for the Dragon — software or hardware — ring the News Desk on 01-437 4343

Discs from Preston

R & A J PRESTON are collating some popular older titles on disc and tape, to be released shortly.

Ruby Robba, *Perilous Pit* and *Desperado Dan* are together on one disc, and *Kung Fu, The Master* is on another. *Space Treck* and the game *Reversi* will also be available together.

Cassette versions will cost £2.99, and disc versions £4.99. For more information contact R & A J Preston, Kings Hall Court, St. Bridges Major, Mid. Glamorgan CF32 0SE, tel. (0656) 880965. Even as I w rite, Preston's are representing Dragon users at the PCW SHow in London.

NDUG see double . . .

Here's the August issue of *Dragon Update*. Paul Grade is beginning to make Editor-on-the-point-of-cracking noises (it's the heat, the dust...). I've noticed that professional editors only go *slightly* off their trollies. Amateur ones go *right* off their trollies. You can't get away from it, you see, except by leaving the country. Which you can't do, because there's too much work.

The August *Update* has articles on Pascal for beginners, easy machine code, running C under DragonDOS, a review of *Music Maker*, some DeltaDOS news (with Dave Martin waxing wroth because more Delta users aren't registering. What, not speaking to each other? United we stand, remember.), a couple of small Tandy hardware reviews, some good advice for people thinking of starting their own magazines (after the demise of *Dragon's Tail*), and a review of Quickbeam's *Galactic Gus* ... which brings me to

Tim Lomas (the editor) sounding off about people sending the same article to *Update* and to other publications. Which usually means *Dragon User*. I only found out about *Galactic Gus* when it was too late to paste something over it. Now, whereas I entirely believe in the author's innocence — it's so difficult to get things printed that it always seems like good sense to run off four copies and send them to different publications — I have to request that anyone else who feels innocent, gets wise. Broad-mindedness is all very well, but actually people don't like buying two different mags and reading the same things in both of them (the occasional reprint is acceptable, but that's all), and if you think about it, writers are pretty lazy, and if we all did that you'd have a choice of eight identical newspapers every morning. So please don't do it. At best you won't get paid, and at worst you will be very unpopular, and have even more difficulty getting things printed.

Mike tells me that he even asked Paul if he minded it being printed elsewhere. This is what I mean by innocence, and in token of that I shall only be donating *half* his fee to *Dragon Update*.

So now that Tim and I have both said it, no-one can say they weren't warned. National Dragon User Group, c/o Paul Grade, 6, Navarino Road, Worthing, Sussex.

OS-9 offer from Denmark

HANS Christian Andersen Computer Inc. have been in touch with us with a special offer for the months of November and December

As regular readers will be aware, HCA are the official licensees for OS-9 for the Dragon, which enjoys a high reputation, but a correspondingly high price.

For November and December, HCA will be offering OS-9 at the considerably lower price of £87.00, as advertised, as well as 20% off the price of other software in their regular advertisements.

You should find further details somewhere in this issue of *Dragon User*, but if you want to contact HCA for details, or order, without further ado, their address is H.C.Andersen Computer Inc., Englandsvej 380, DK-2770 Kastrup, Denmark. Phone 01 52 44 04. Postage for software is 5% of the order price, up to £5.

London Show

John and Helen Penn of John Penn Discount Software are organising the 6th 6809 Dragon Show in London on Saturday 5th December.

The show is to be held in the Great Hall of the Connaught Rooms from 10am to 4pm, according to our present information, and we'll be bringing you more information when we receive it. The Connaught Rooms are altogether a warmer and more hospitable venue than the Horticultural Halls, and the Penns are organising guest demonstrators as well as trade stands.

This is a good opportunity for everyone who lives within striking distance of The Smoke to come and support one of the Dragon's most loyal suppliers over the years, as well as many other Dragon names, and do a bit of shopping in time for Christmas. The future of the 6809 Show, and indeed all Dragon shows, depends on users turning up to see what's doing, even if they don't plan to spend a fortune or buy every new product on the market. And its a great way to meet other Dragoners as well.

You might even get a laugh by watching the staff of *Dragon User* recite the ballad of the September Issue another few hundred times, with themes and variations.

New Starship games

STARSHIP Software were at Rochdale, demonstrating all their new software and the old stuff as well. We have *Rola-ball* in the reviews pipeline. *Rola-ball* is described as '49 screens of 3D magic! Guide the ball over a hazardous 3D landscape, collecting jigsaw pieces en route. Beware the cube and the sheer drops!'

Also on the horizen are *CAD-6809*, with 'lines, boxes, circles, discs . . . ZOOM command for pixel perfect pictures, many types of brushes for freehand mode. As used to create all Starship title pages,' and *Windos*, with '10 user definable windows, icons, upll-down menus, and a pointer, a 32 x 24 display, a real time clock, plus much more! Complete with a suite of dedicated utilities. Easily incorporated into your Basic programs. Does NOT require 64K.' No price has been announced for the new programs as yet, but Starship's programs typically retail at around £3 to £5.

Starship's other titles are the *Destiny* series and *Composer Componion* and *El Pea*, a collection of ragtime and jazz tunes written on *Composer Companion*, all published by Microvision, and *Hi-Res Text*, available from John Penn.

More room in the Arc

ARC Software are reducing the price of their adventure game *The Thirteenth Task* from £3.00 to £1.99 (including p&p) as a special Christmas discount from 1 December 1987 to 31 January 1988.

The Thirteenth Task would be better known if Arc were in a position to advertise more freely, and is well spoken of by those who have played it. An adventure for the price of a discount game is definitely a bargain.

Contact Arc Software at 272 Mearns Road, Newton Mearns, Glasgow G77 5LY.

Show in Cardiff

SPONSORED by Dragon dealers and general wholesalers R & A J Preston, the Wales and West Computer Show in November will feature the Dragon as well as other computers. The show is to be held on 21 November at the Central Hotel in Cardiff from 10am to 4pm, and the entrance fee is only £1.

For further information can contact R & A J Preston on Southerndown (0656) 880965.

Dragonsoft

Save and lose

Program: *Telewriter* Delta disc upgrade
Supplier: Microdeal
Price: TBA

THE instruction manual which comes with Microdeal's *Telewriter* is quite simply the best I have ever seen. So far so good, but it's time for another gripe, because I'm afraid I want to talk about the Delta disc upgrade of *Telewriter*. My copy had the most incredible bug in it. I thought it was just me, until I checked around, and lo and behold, someone else had the same problem, which is simply this: if you've used the printer, it is 100% certain that when you next access the disc I/O the program is going to go into an irreconcilable crash. If you don't print your work before saving it, then there's no certaintly of a crash, just a very big chance. So far, since buying this upgrade, I have lost two acts of a play, certain copy for this magazine, a short story, the first eight chapters of a book, and about fifty assorted songs and poems. At this rate I'm going to follow Wayne Smithson's example and switch to an Amstrad! Microdeal should stop selling this package until the bug has been sorted out.

Jason Orbaum

(See also this month's Letters page. Editors usually give writers about three days' grace to get over a system crash, so if your WP does even once what Jason's WP seems to have done several times, do the sensible thing — don't trust it with anything longer than half a page at a time. — Ed.)

Four in one

Name: *Airball*
Supplier: Microdeal
Price: £5.95

Name: *Stone Raider II*
Supplier: Microdeal
Price: £5.95

Name: *Trun*
Supplier: R & A.J. Preston
Price: £3.99

Name: *Adventure Writer*
Supplier: Cowan Software
Price: £14.95

MICRODEAL have recently come out with *Stone Raider II* and *Airball*. Of the two, *Airball* is definitely the better designed although it is very, very hard. The game puts you in a castle dungeon beautifully drawn in black and white 3-D. The idea is to find and return a book of spells that the wizard has lost. It's basically a mapping game but mapping is not the key to the game; the key to success lies in negotiating your way around the screens without hitting the spikes, and getting to the pumps before you run out of air. The game is really beautiful to play and involves a lot of skill and fast perception. It is a great game on joystick or keyboard, unlike *Stone Raider II* which is a waste for one simple reason — there is no keyboard option and the game is too fast for joystick work.

In a game like *Boulder Crash* (which is what this is), where positioning and timing are far more important than speed, dicey play control like this is farcical. The game contains a cheat for 'those who can find it' which is becoming a general trend days, much to my distress. I have like to be able to buy a game that I don't have to work out how to hack to get the full enjoyment from it. If a program has a screen designer in it I would like to use it without having to solve several convoluted problems — if I want to do this I'll buy an adventure game.

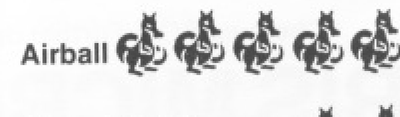
However, that fault taken into account there are some very interesting layouts in *Stone Raider II* and it seems to have been well thought out. The sound isn't anywhere near Chris Jolly standard but it gets by and the graphics are quite attractive. I've you've got *Boulder Crash*, and have finished it with ease, then this one is for you as it really starts where that one finished, but why oh why, this late into the Dragon's life, are we getting games without keyboard control??

Moving on we come to *Trun*, one of Blaby's last releases, which has all the elements of the arcade game in it, but all of them badly ripped off. In the arcades this was one of my favourite games (and, due to its excellent tongue in cheek sense of humour, one of my favourite films) but this game just doesn't do it justice. The cycles are embarrassingly easy, the thanks are impossibly hard, the spiders are messy and the cone is the sort of graphics my two year old cousin would design! Basically, this version of *Tron* is no better than the other really useless one that's been on the market for quite some time now.

Adventure Writer from Cowen software is, I'm sure a superb piece of programming and I'm equally sure that its creator has written many very good adventures on it. My copy, however, has photocopied A4 sheet after photocopied A4 sheet of comprehensive instructions none of which have helped me even to start to use the program! It's supposed to be an adventure writing language, well if that's true then it's the most user unfriendly language I've ever used! It makes FORTH seem like baby talk and assembler seem like BASIC. I would really like to be able to give this program a good review because I think a utility of this nature is quite exceptionally useful. However, if it can't be used because someone didn't take time to test out the instructions then what's the point?

Before I sign off and give my ratings on these programs, please do not think I am being pessimistic. I love the Dragon, it is the only computer I own, and it is still the best programmers' machine on the market. This is an appeal to the software companies. If you can get games like *Airball* and *Superkid* then give them to us. We're fed up with trying to sort through the chaff to find the wheat! Oh yes, and one other thing, whoever is selling Mastertronic's *Vegas Jackpot*, please stop it . . . you're embarrassing me!!

Jason Orbaum

Airball 

Stone Raider

Adventure Writer 

Trun

High life

Program: *The Immortal Strain*
Supplier: Broomsoft
30 Broomhill, Welwyn Garden City, Herts HL7 1RF
Price: £4.00

ITS time to bring out those old adventure favourites of assorted metal rings, men in coffins and devil statues in Michael Edwards' latest Broomsoft adventure.

As for your purpose in collecting such objects, well according to *Dragon User* news desk a few months back it's in an attempt to become immortal (that explains the title). I revert to this old news item as there are no initial instructions anywhere. This may well be a secret mission, but they could at least inform the poor person who has to do it!

Actually lack of instructions doesn't prove a problem because as with any adventure you set off to wander around and pick up any useful looking objects. Also there's a very handily discarded note in a church telling you about various items to help you succeed in your test.

Your task begins in a pentacle where there's a 'feeling of magic' according to the redesigned white text on black background. I would not disagree with this description as when you drop specific objects within its boundary your score mysteriously rises. There is a score feature, by the way, so you can check on your progress throughout the game.

One of your initial problems is a not very nice fellow called Zargron, whose mysterious name aptly fits the atmosphere generated in the program. This nasty chap keeps following you about, and then for no other reason than he's a bit of a madman (or mad thing, whichever is applicable) gives you immortality rather earlier than you would expect. Except that sitting on a fluffy white cloud playing a harp is not exactly what you were after.

A location called 'the old green' soon provides a method of despatching Zargron. Having done to him as he would undoubtably have done to you, the old cad then resurrects himself for a few seconds to try and get his hands on a nearby amulet. Not exactly cricket — but then again you're not quite

at the MCC, as shown by the fact that he leaves a map written in Swahili.

Wandering in peace around beaches, cliffs, forests and a church for good measure should reveal to you rings of different tinctures, one found by the old graveyard and coffin scenario, when the text actually satinises itself.

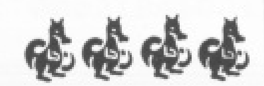
The chance to go up and down as well as traditional compass directions is also used. By exploring downwards you could find yourself amid a sea of fire with a hand stretching out of the inferno.

Also by diverting down you can get on a beach where further along the shore is a motorboat willing to transport you to three other areas (and to the main shore, although as you're initially there it seems a bit of a waste of petrol). In theory the boat should take you to these locations but it only seems to want to go to the small islet and back to the main shore. Southern and northern waters you are told have nothing of interest. If there's nothing worth acquiring there why have a motor boat going to them? Sounds about as truthful as a sentence from a second hand car salesman methinks!

Conclusion time is approaching and therefore I'll have to try and sum the game up. It doesn't seem over difficult as the problems are logically solved, but what do you throw the grenade at and why does the computer reply 'sorry' when I try to get into the tent? (No doubt I shall solve these problems about two minutes after I've posted this review.) The game through the disused church and graveyards has an ominously eerie atmosphere about it and is as well written as any text adventure by a big software house for the Dragon.

Big software houses brings me neatly onto my parting shot. Just because this game wasn't advertised in pages of glossy blurb it doesn't mean its not worth buying. Small firms like Broomsoft deserve all the support they can get — they're not after vast profits but just providing the public with adventures, and good ones at that!

Philip Stott



Cream cake quaffer

Program: *Catacomb Crisis*
Supplier: Microvision Software
Price: £7.95

MICROVISION'S adverts for this title have been describing it as a 'graphic adventure'. On playing through it becomes apparent that it is only just bordering on being an 'arcade adventure'.

This slight misrepresentation doesn't of course alter its quality, but traditional adventure freaks might like to know that the only full word you type in is your name and that movement is totally by joystick.

Setting off with a grumble may appear as if I don't like the game, but the opposite is true. Your objective as with all games is to save your neck, in this case by escaping the numerous vast labyrinths of Tyros mountain. How and why you're there in the first place is irrelevant; you just have the problem of getting out intact.

In the caverns you are shown an overhead 2D view, and as its pretty black (or red in this case) in underground caves you have a lamp which lights up the walls with a green glow. The catacomb itself is made up of a series of simply drawn passageways interspersed with larger caves where various nasty creatures and also more useful objects are found.

Only a small section of the whole network is shown on the screen and therefore you are able to call up a map which shows all that has so far been discovered. By using this map it becomes apparent that it's not merely a question of learning where everything is on a level and then completing it almost without thinking the next time as a new network is created for every game, adding much more variety.

As well as showing where you are the main screen has a box for displaying messages and commands, and a score panel revealing your status and standing.

To pick something up you just move over the object and can then use with various single letter commands. For instance, if you find a dagger to hold the weapon you type 'H' and then it corresponding letter on the inventory, ie the third object on the list is letter C and you therefore enter that.

As well as holding weapons threateningly you can put on or take off clothes or armour, quaff potions, read scrolls, throw objects around and eat. Edible objects are normally apples, cherries or even power pills, though here they're cream cakes — whether you're trapped in a cavern or not you're obviously not on a diet!

You don't only have to eat to survive but to keep your hit points up. How many hit points you have determines how long you can stand the attacks of 'globs' 'sparks' and other monsters.

The more monsters that are killed the higher the experience level and the greater the limit on the number of hit points. The hit point level can be replenished by staying out of attack mode. But this mode obviously allows you to attack your foes and also stops them giving you a fair old thrashing.

There are also bags of gold scattered about for good measure, although they don't look much like money filled sacks,just like the cakes don't look like cakes. None of the graphics are razor sharp like, say *Airball*, although this is partly due to the fact that the author hasn't sacrificed his four colours just for the sake of accuracy.

There is of course the traditional high score table, determined by gold not experience or cavern level as perhaps would be expected. As for music, well there isn't any. The author obviously isn't a composer as the only sound your Dragon utters is the thud of heavy feet padding up and down the passageways.

As with all good games you want to get that bit further and eventually hopefully manage to complete it. With progression you gradually find more potions to use, tougher armour and weapons with greater power, and as always more monsters — how easy life would be without them!

I must be starting to get reasonable at this game as I'm now playing games which are beginning to take over the hour rather than over the minute. If you haven't got a couple of hours for one game then there's the old adventuring option of saving on to a handy blank tape and resuming later on — well perhaps its not so unlike an adventure after all.

Philip Stott



Bargain screens

Program: *Frankie*
Supplier: Quickbeam
Price: £6.95

WHEN Wayne Smithson started writing this it was as an alternative to Roy Coates's 100 screen game *Talisman* which never made it to the market so this is the ONLY 100 screen game around. As well as being a one-of-a-kind game it has the only loading systems of its type which scrolls the instructions across the middle of the screen while it loads, and the game takes about the same length of time to load as the average Dragon game.

The game can be in one of two colour sets: mode3, screenl,0 or pmode4, screen1,1. You can use joystick or keyboard and there is of course a 'reserved' section which is called Hacker's Delight, which has been mentioned in The Expert's column.

There is an option to redesign the screens in which you can save and load the screens.

On to the game. You are Frankie Stein (sick, I know) who has to collect all the keys on the screen in order to go on to the next screen. Trying hard to stop you are eyes, feet, noses, mouths, hands, vampire bats and even Egor, sent to fire lightning bolts at you. Thankfully you have a defence against this lot, a laser, but when you use it your energy goes down and when your energy gets to 0 you lose a life. To top it all you have collapsing platforms, killer pads and conveyer belts. Thank heavens for Hacker's Delight!

I can't give *Frankie* five out of five, as it isn't quite addictive enough, but it's definitely worth four at the price.

Stephen Cogan



Down in the dumps

***Dragon User** frequently gets requests for screen dumps. Here we present some specimens from our collection*

Tandy CGP-115

This program is a four colour screen dump for the GCP-115. After numerous attempts I have finally managed to get a screen dump of reasonable size in two hours instead of 8) hours.

The speed of this program is mainly due to the STEP 2 in line 90. The reason for this is that each point on the PMODE 3 graphics screen is made up of two pixels so only half the number of pixels have to be sampled, which cuts down dumping time. As the paper in the CGP-115 is only 4) inches wide, the screen will be dumped on its side.

If there are any problems I will sort them out if you send your question with a stamped self-addressed envelop to *Dragon User* for forwarding. *P. Marlow*

Brother M1009

THE following machine code listing is a program which contains two different sized screen dumps. The first one dumps the screen to the printer four times bigger than if done one dot for one pixel. The second is nearly the same as the first, except the screen is printed sideways; the final dump it the same width but is taller.

The program was originally written for dumping to a Brother M1009 printer but should be able to dump to Epson and printers which use a similar code.

To load, type in **listing 1**, the hex loader, and run it. Enter the data from **listing 2**, entering ten bytes of data (1 line) at a time. After entering the single string, enter the checksum. If there are any errors you will be asked to enter the line again. Once all the data has been entered, save it by typing:

```
CSAVEM "DUMPS",
  &H4E21,%H4FED,&H4E21
```

To run the first screen dump, type

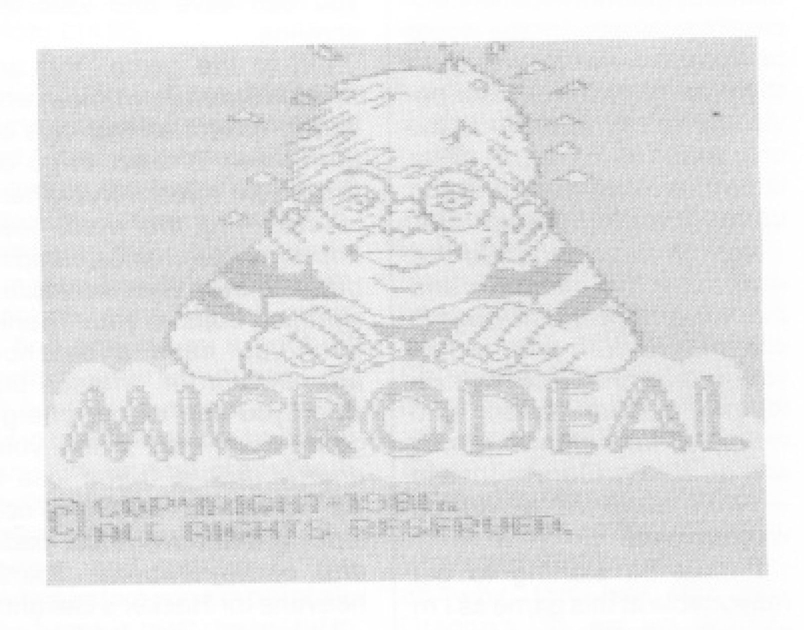
```
EXEC &H4E211
```

and to run the second, type

```
EXEC &H4EFA
```

D. Diggins

Tandy CGP-115




```
10 REM SCRNDUMP MK12
20 REM 4 COLOUR SCREEN DUMP FOR THE CGP-
115 PRINTER
30 REM (C) P. MARLOW 20/8/87
40 REM EACH DUMP TAKES APPROX. 2 HOURS
50 PMODE 3,1:SCREEN 1,0
60 CLOADM
70 PRINT #-2,CHR$(18):PRINT #-2,"M432,0"
:PRINT #-2,"I"
80 FOR Y=0 TO 191
90 FOR X=0 TO 255 STEP 2
100 P=PPOINT(X,Y)
110 IF P=2 THEN PRINT #-2,"R0,-4":GOTO 1
40
120 PRINT #-2,"C";P-1
130 PRINT #-2,"J0,-4"
140 NEXT X
150 M=M-2:PRINT #-2,"M"+STR$(M)+",0"
160 NEXT Y
170 PRINT #-2,CHR$(17):END
```

Brother M1009

```
LISTING 1 - HEX LOADER

10 CLEAR200,20000
20 FOR X=&H4E21 TO &H4FE3 STEP 10
30 PRINTHEX$(X);":"
40 T=0
50 INPUT A$
60 L=LEN(A$)
70 IF L<>20 THEN PRINT"error-enter line again"
   :GOTO30
80 FOR Y=1 TO LEN(A$) STEP 2
90 P$=MID$(A$,Y,2)
100 P=VAL("&H"+P$)
110 POKE X+((Y-1)/2),P
120 T=T+P
130 NEXTY
140 INPUT"= ";C$
150 IF VAL("&H"+C$)<>T THEN L=19:GOTO70
160 NEXT X
```

```
LISTING 2 - MACHINE CODE

4E21:  861BBDBCF58640BDBCF5 =643
4E2B:  860ABDBCF58E0600108E =430
4E35:  4FDCC609A6A0BDBCF55A =608
4E3F:  26F8861FB74FD94FC680 =537
4E49:  F74FDA108E0007E68453 =482
4E53:  F44FDA265FE68453F44F =5A2
4E5D:  DA265BE6882053F44FDA =559
4E67:  2656E6882053F44FDA26 =4A0
4E71:  51E6884053F44FDA264C =4E1
4E7B:  E6884053F44FDA2647E6 =571
4E85:  886053F44FDA2642E688 =52E
4E8F:  6053F44FDA263DBDBCF5 =5A1
4E99:  BDBCF5BD800681511027 =4BA
4EA3:  0054108C0000272C313F =1B3
4EAD:  B64FDA46B74FDA4F2099 =50D
4EB7:  8B80209D8B4020A18B20 =3FF
4EC1:  20A68B1020AB8B0820B0 =38F
4ECB:  8B0420B58B0220BA8B01 =357
4ED5:  20BFB64FD98100270E4A =3BD
4EDF:  B74FD98680B74FDA3001 =4F6
4EE9:  16FF5A8C1D8010240006 =2D2
4EF3:  30886116FF3A39861BBD =3FF
4EFD:  BCF58640BDBCF5860ABD =632
4F07:  BCF51700BE8602B74FDB =4EF
4F11:  8E1DE0108E00C0BD8006 =42C
4F1B:  81511027FFD88C1E0010 =39A
4F25:  24FFD1B64FDB8102273B =4B9
4F2F:  2068BDBCF5BDBCF53088 =61C
4F39:  E0313F26DAB64FDB4AB7 =531
4F43:  4FDB26148602B74FDB30 =3FD
4F4D:  891801860ABDBCF51700 =3B7
4F57:  7216FFB930891800860A =3A1
4F61:  BDBCF517006316FFAAE6 =58D
4F6B:  844F53C4802617E68453 =464
4F75:  C4402614E68453C42026 =405
4F7F:  11E68453C410260E20A8 =39E
4F89:  8BC020E58B3020E88B0C =4AA
4F93:  20EB8B032098E6844F53 =45D
4F9D:  C4082618E68453C40426 =3B5
4FA7:  15E68453C4022612E684 =43A
4FB1:  53C401260F16FF788BC0 =425
4FBB:  20E48B3020E78B0C20EA =467
4FC5:  8B0316FF67108E4FE5C6 =4A2
4FCF:  08A6A0BDBCF55A26F839 =56D
4FD9:  1620021B41080A1B2A05 =F0
4FE3:  00021B41081B2A008001 =12C
```

```
4E21                  ************************
4E21                  *  LARGE SIZE DUMP   *
4E21                  ************************
4E21
4E21 861B         LARGE  LDA    £27
4E23 BDBCF5              JSR    48373
4E26 8640                LDA    £64
4E28 BDBCF5              JSR    48373
4E2B 860A                LDA    £10
4E2D BDBCF5              JSR    48373
4E30 8E0600       DUMP   LDX    £1536
4E33 108E4FDC     START3 LDY    £CODES1
4E37 C609                LDB    £9
4E39 A6A0         GETCOD LDA    ,Y+
4E3B BDBCF5       SENCOD JSR    48373
4E3E 5A                  DECB
4E3F 26F8                BNE    GETCOD
4E41 861F         START2 LDA    £31
4E43 B74FD9              STA    ROW
4E46 4F           START  CLRA
4E47 C680                LDB    £128
4E49 F74FDA              STB    NUM
4E4C 108E0007            LDY    £7
4E50 E684         LOOP1  LDB    ,X
4E52 53                  COMB
4E53 F44FDA              ANDB   NUM
4E56 265F                BNE    T1
4E58 E684         N2     LDB    ,X
4E5A 53                  COMB
4E5B F44FDA              ANDB   NUM
4E5E 265B                BNE    T2
4E60 E68820       N3     LDB    32,X
4E63 53                  COMB
4E64 F44FDA              ANDB   NUM
4E67 2656                BNE    T3
4E69 E68820       N4     LDB    32,X
4E6C 53                  COMB
4E6D F44FDA              ANDB   NUM
4E70 2651                BNE    T4
4E72 E68840       N5     LDB    64,X
4E75 53                  COMB
4E76 F44FDA              ANDB   NUM
4E79 264C                BNE    T5
4E7B E68840       N6     LDB    64,X
4E7E 53                  COMB
4E7F F44FDA              ANDB   NUM
4E82 2647                BNE    T6
4E84 E68860       N7     LDB    96,X
4E87 53                  COMB
4E88 F44FDA              ANDB   NUM
4E8B 2642                BNE    T7
4E8D E68860       N8     LDB    96,X
4E90 53                  COMB
4E91 F44FDA              ANDB   NUM
4E94 263D                BNE    T8
4E96
4E96 BDBCF5       PRINT  JSR    48373
4E99 BDBCF5              JSR    48373
4E9C BD8006       KEY1   JSR    $8006
4E9F 8151                CMPA   £81
4EA1 10270054            LBEQ   BASIC
4EA5 108C0000            CMPY   £0
4EA9 272C                BEQ    END
4EAB 313F                LEAY   -1,Y
4EAD B64FDA              LDA    NUM
4EB0 46                  RORA
4EB1 B74FDA              STA    NUM
4EB4 4F                  CLRA
4EB5 2099                BRA    LOOP1
4EB7
4EB7 8B80         T1     ADDA   £128
4EB9 209D                BRA    N2
4EBB 8B40         T2     ADDA   £64
4EBD 20A1                BRA    N3
4EBF 8B20         T3     ADDA   £32
4EC1 20A6                BRA    N4
```

```
4EC3 8B10          T4      ADDA  £16
4EC5 20AB                  BRA   N5
4EC7 8B08          T5      ADDA  £8
4EC9 20B0                  BRA   N6
4ECB 8B04          T6      ADDA  £4
4ECD 20B5                  BRA   N7
4ECF 8B02          T7      ADDA  £2
4ED1 20BA                  BRA   N8
4ED3 8B01          T8      ADDA  £1
4ED5 20BF                  BRA   PRINT
4ED7 B64FD9        END     LDA   ROW
4EDA 8100                  CMPA  £0
4EDC 270E                  BEQ   RET
4EDE 4A                    DECA
4EDF B74FD9                STA   ROW
4EE2 8680                  LDA   £128
4EE4 B74FDA                STA   NUM
4EE7 3001                  LEAX  1,X
4EE9 16FF5A                LBRA   START
4EEC 8C1D80        RET     CMPX  £7552
4EEF 10240006              LBHS   BASIC
4EF3 308861                LEAX  97,X
4EF6 16FF3A                LBRA   START3
4EF9 39            BASIC   RTS
4EFA               ********************
4EFA               * EXTRA LARGE DUMP *
4EFA               ********************
4EFA 861B          XLARGE LDA  £27
4EFC BDBCF5                JSR   48373
4EFF 8640                  LDA   £64
4F01 BDBCF5                JSR   48373
4F04 860A                  LDA   £10
4F06 BDBCF5                JSR   48373
4F09 1700BE                LBSR  PTA
4F0C 8602                  LDA   £2
4F0E B74FDB                STA   ROU
4F11 8E1DE0                LDX   £7648
4F14 108E00C0      L1A     LDY   £192
4F18 BD8006        L3A     JSR   $8006
4F1B 8151                  CMPA  £81
4F1D 1027FFD8              LBEQ  BASIC
4F21 8C1E00                CMPX  £7680
4F24 1024FFD1              LBHS  BASIC
4F28 B64FDB                LDA   ROU
4F2B 8102                  CMPA  £2
4F2D 273B                  BEQ   FIR
4F2F 2068                  BRA   SEC
4F31 BDBCF5        L2A     JSR   48373
4F34 BDBCF5                JSR   48373
4F37 3088E0                LEAX  -32,X
4F3A 313F                  LEAY  -1,Y
4F3C 26DA                  BNE   L3A
4F3E B64FDB                LDA   ROU
4F41 4A                    DECA
4F42 B74FDB                STA   ROU
4F45 2614                  BNE   L4A
4F47 8602                  LDA   £2
4F49 B74FDB                STA   ROU
4F4C 30891801              LEAX  6145,X
4F50 860A                  LDA   £10
4F52 BDBCF5                JSR   48373
4F55 170072                LBSR  PTA
4F58 16FFB9                LBRA  L1A
4F5B 30891800      L4A     LEAX  6144,X
4F5F 860A                  LDA   £10
4F61 BDBCF5                JSR   48373
4F64 170063                LBSR  PTA .
4F67 16FFAA                LBRA  L1A
4F6A E684          FIR     LDB   ,X
4F6C 4F                    CLRA
4F6D 53                    COMB
4F6E C480                  ANDB  £128
4F70 2617                  BNE   F1
4F72 E684          Q2      LDB   ,X
4F74 53                    COMB
4F75 C440                  ANDB  £64
4F77 2614                  BNE   F2
4F79 E684          Q3      LDB   ,X
4F7B 53                    COMB
4F7C C420                  ANDB  £32
4F7E 2611                  BNE   F3
4F80 E684          Q4      LDB   ,X
4F82 53                    COMB
4F83 C410                  ANDB  £16
4F85 260E                  BNE   F4
4F87 20A8                  BRA   L2A
4F89 8BC0          F1      ADDA  £192
4F8B 20E5                  BRA   Q2
4F8D 8B30          F2      ADDA  £48
4F8F 20E8                  BRA   Q3
4F91 8B0C          F3      ADDA  £12
4F93 20EB                  BRA   Q4
4F95 8B03          F4      ADDA  £3
4F97 2098                  BRA   L2A
4F99 E684          SEC     LDB   ,X
4F9B 4F                    CLRA
4F9C 53                    COMB
4F9D C408                  ANDB  £8
4F9F 2618                  BNE   D1
4FA1 E684          S2      LDB   ,X
4FA3 53                    COMB
4FA4 C404                  ANDB  £4
4FA6 2615                  BNE   D2
4FA8 E684          S3      LDB   ,X
4FAA 53                    COMB
4FAB C402                  ANDB  £2
4FAD 2612                  BNE   D3
4FAF E684          S4      LDB   ,X
4FB1 53                    COMB
4FB2 C401                  ANDB  £1
4FB4 260F                  BNE   D4
4FB6 16FF78                LBRA   L2A
4FB9 8BC0          D1      ADDA  £192
4FBB 20E4                  BRA   S2
4FBD 8B30          D2      ADDA  £48
4FBF 20E7                  BRA   S3
4FC1 8B0C          D3      ADDA  £12
4FC3 20EA                  BRA   S4
4FC5 8B03          D4      ADDA  £3
4FC7 16FF67                LBRA  L2A
4FCA 108E4FE5      PTA     LDY   £CODES2
4FCE C608                  LDB   £8
4FD0 A6A0          GETCO   LDA   ,Y+
4FD2 BDBCF5        SENPR   JSR   48373
4FD5 5A                    DECB
4FD6 26F8                  BNE   GETCO
4FD8 39                    RTS
4FD9 00            ROW     FCC   0
4FDA 00            NUM     FCC   0
4FDB 00            ROU     FCC   0
4FDC 1B41080A18 CODES1 FCC   27,65,8,10,27,42
4FE2 050002                FCC   5,0,2
4FE5 1B41081B2A CODES2 FCC   27,65,8,27,42,0
4FEB 8001                  FCC   128,1
4FED
```

Pamcodes

Part one of a new series on machine code for beginners by **Pam D'Arcy.**

HAVING volunteered to try and help R.F. Harding (*Letters*, September issue) and others like him, I have a couple of days before deadline and just one page to do, so you can imagine what you are going to get this month — a lot of waffle and no real detail! However, because the 'getting started' syndrome described in R.F. Harding's letter is often a big problem, I have included a few lines of assembler source code to see if we can all get going without wasting another month simply waiting for the first real article.

If you are trying to learn Dragon machine code for whatever reason, I will assume that you will already have

1) at least one book on the subject
2) some form of assembler software

Rather than dive in with a blow by blow account of machine code, which may of necessity leave you dangling mid-code from one long month to another, my intention will be to present a series of self-contained articles. Some of them will refer to programs from popular Dragon machine code books that people have had problems with, so that those of you who have those books in your possession, or want to get them, can go back later and tackle them with more confidence. This also opens the floodgates for you to swamp me with examples you find difficult, which I may be able to write an article about.

Lack of standards

Using the Dragon's inbuilt Basic, variables may be numeric or string, and are given alphanumeric names of any length — except that for various planned and very good reasons, these names are truncated within the Dragon to the first two characters when the Basic program is RUN. Thus variables names, for examples, QUORUM and QUANTITY would both be treated as a variable named QU, and the same memory location would be used every time either variable QUORUM or QUANtity (or any other variable commencing with the letters QU) was encountered when RUNning a Basic program — which could account for some unexpected results...

```
10 QUORUM=10
20 INPUT"MEMBERS
   PRESENT";QUANTITY
30 IF QUANTITY<QUORUM THEN
   ?"NO QUORUM" ELSE
   ?"COMMENCE MEETING"
```

(That is as I would enter it — I never type in the word PRINT but always use the available shorthand question mark form).

In a similar manner, Basic commands, or keywords, are replaced internally by a value, or token, that occupies a single byte of memory as the line of code is entered into the program (a procedure known as crunching — and decrunching when it is converted back for LISTing purposes). If any of the names of our variables commence with the same letters as a keyword, the CRUNCH routine positively ruins our programming intention ...

```
100 TOTAL=BALANCE+VALUE
```

It lists correctly because of the DECRUNCH routine, but when attempting to RUN the program, TO of total and VAL of value will be tokenised keywords that the Basic rightly says do not make sense, so it gives a syntax error.

Why have I taken this time out to go over what you probably know already about Dragon Basic when it is machine code that you are keen to get to grips with? In the same way that Dragon's Basic (and all other Basics) has its own quirks and rules, so do assemblers — and unfortunately there is no one inbuilt assembler on the Dragon. Assemblers are simply a means to an end, a program that converts source code written according to ITS OWN rules, to object or machine code that it places (again, depending on its own rules) at its own or user specified addresses in memory. Apart from the fact that in general the 59 different types of source code instructions are represented by standard groups of letters, or mnemonics (memory joggers, as in SUB for SUBtract and BNE for Branch Not Equal) and that preceding the mnemonic field in a line of source code will be a label field, and following the mnemonic will be an operand field, there any similarities between assemblers will end, making it impossible for anyone to present source code in a form suitable for input direct into all (or probably even most) assemblers produced for the Dragon. I shall therefore keep the code as straightforward as possible so that minimum adjustment is needed for it to be immediately acceptable to your assembler.

To rescue your Basic

I'll now include a few lines of source code for you to play with. There is no room to explain anything about this routine or its machine code this month, other than to say that once it has assembled correctly (perhaps save it to tape or disc at this point, too), if you type in a short Basic program (10 REM will do); LIST it; type in NEW then LIST again — lo, the Basic program has gone. EXEC the assembled routine from its start address and at OK, LIST again and the Basic program will have been restored to its former glory. It is also useful should you have an I/O error on your only saved copy of a Basic program. Load in this machine code routine and EXEC it, and the program lines as far as successfully read will be available to you. This routine can be found in *Dragon User* July 1983, page 44 and June 1984, page 29, and in the *Dragon Programmers Reference Guide* by John Vander Reyden (Melbourne House), pages 107 — 108. Note the printing errors in the assembler listing on page 108 if you had trouble with it — a common enough problem that I shall deal with another time. The book is still available from Peaksoft.

Listing 1: To rescue a 'lost' Basic program, 32k mode

Expected generated object code	Source code			
	Line number	Label	Mnemonic	Operand
9E 19	10	GO	LDX	$19
BD 83 F3	20		JSR	$83F3
30 02	30		LEAX	2,X
9F 1B	40		STX	$1B
9F 1D	50		STX	$1D
9F 1F	60		STX	$1F
39	70		RTS	
	80			

Assembly

There are two stages to assembly: source code input, or editing, and the assemble process itself, or generation of object or machine code. Your assembler may include its own editing facilities, or it may assemble source code statements entered using the Basic editor. Enter the source code according to your assembler's instructions and rules. Line numbers may or may not be needed. Using them when not required or vice versa is likely to cause every line to be in error. The label on line 10 is not needed by the program itself, but is included in case it is mandatory in your assembler that the first instruction will be labelled. Your assembler may need an @ or other symbol to procede the label name. It may need the source code to be entered in particular columns of a line or fields amy need to be separated by a specific character. The last line of source code may need a special value in the label or mnemonic field, hence my leaving a line number 80 there, to remind you.

The assembler instructions should tell you where the object code will be placed in memory, which may depend on any CLEAR statement you type in before loading or running the assembler. The above program will run successfully from any address within the Dragon provided, obviously, that it is not loaded into areas occupied by the Basic program that is being recovered, nor into Basic's workspace. (Reserved graphics pages and higher addressed memory reserved for machine code by using an appropriate CLEAR statement are idea areas). With routines like this (that uses a routine within the 32K Basic ROM, line 20) Dragon 64 owners who would want to use the routine in 64K mode will need to add $4000 to the operand

```
JSR $C3F3
```

with correspondingly different object code being produced.

Motorbiking

***Richard Boryna** catches his busses in arcade style.*

THIS is a version of the arcade game where you control an intrepid biker as he leaps over buses. You start with four motorbikes and five buses to jump, building up to twelve if you can manage it. To land successfully your back wheel must hit the top two-thirds of the down ramp. Bikes are wrecked by jumping too far, not far enough or going too fast and flipping the bike over.

The right-hand joystick port is used and an analogue joystick is needed to give proportional control. When it is centred you will have a slight forward velocity. Pull back to slow or stop and push forwards to accelerate. To start your run up, press the fire button.

The engine noise is interrupt driven by re-routing the IRQ interrupt by altering the address at locations 269/270. This has to be done by loading the new address from cassette while the interrupt is disabled, as altering it from inside the program can have disasterous results. This is why the first time the game is run the cassette recorder 'play' button must be left down.

To enter the program type in the BASIC listing and test it, first making line 300 a REM statement. When it is running properly and the data in lines 160-190 has been checked, remove the REM from line 300 and CSAVE the program. You will not have sound this time.

Once done, enter the following:

POKE 1769,127
POKE 1770,163

Then, with the tape directly after the main program, enter:

CSAVEM"",1769,1770,1769

The game can then be re-loaded and run, this time with sound. Before loading anything else, turn the computer off as the engine noise routine will corrupt any other sound such as PLAY or SOUND commands.

If you can't face typing the listing in, £2.75 to me at 71 Oakley Drive, Wellingborough, Northants, NN8 3JY will get you a couple of copies on tape.

```
10 ' ***************
20 '****************
30 '** STUNT BIKER **
40 '**  R. BORYNA  **
50 '**  16/5/1987  **
60 '****************
70 ' ***************
80 CLS
90 PRINT@234,"PLEASE WAIT"
100 PCLEAR8
110 '***************
120 '** LOAD CODE **
130 '***************
140 CLEAR300,32674
150 FORI=0TO91:READA$:POKE32675+I,VAL("&H"+A$):NEXT
160 DATA 34,02,B6,7F,E7,8B,01,B7,7F,E7,B1,7F,E4,27,28,B1,7F,E3,27
170 DATA 18,B1,7F,E2,24,05,35,02,7E,9D,3D,B6,7F,E5,B7,FF,20,7F,7F
180 DATA E7,35,02,7E,9D,3D,B6,7F,E6,B7,FF,20,35,02,7E,9D,3D,7F,FF,20,35,02,7E,9D,3D
190 DATA 03,02,01,78,3C,00,00 .
200 DATA BD,8B,27,DD,2B,9E,19,BD,84,03,25,05,30,1F,9F,33,39,C6,0E,7E,83,44
210 DEFUSR0=32745
220 DIMB(2,2),B1(2,2),B2(2,2),B3(2,2),B4(2,2),B5(2,2),B6(2,2),B7(5,5),B8(2,2),B9(2,2),BL(2,2)
230 BN=4:DIMX(5),Y(5):XX=USR0(3640):L=3
240 XX=USR0(3440)
250 AF=25*75/18.45:RD=3.142/180:CA=(COS(RD*45))^2:S2=SIN(RD*45*2):TA=TAN(RD*45)
260 POKE&HFF1D,&H34:POKE&HFF1F,&H35
270 '************************
280 '** LOAD IRQ RE-ROUTE? **
290 '************************
300 IFPEEK(269)<>127 THENCLOADM"",65536-1500
310 PMODE4,5:PCLS:GOSUB2800:GOSUB2440
320 '*************************
330 '** BUSES, RAMP + LIVES **
340 '*************************
350 BM=BN*11
360 FORI=168-BM TO 168 STEP11
370 PUT(I+1,176)-(I+10,190),B7,PSET
380 NEXT
390 LINE(167-BM,191)-(167-BM,176),PSET:LINE-(167-BM-29,191),PSET:PAINT(165-BM,190)
400 BM=168-BM
410 IFL=0 THEN440 ELSEFORI=5TO5+((L-1)*6) STEP6
420 PUT(240,I)-(247,I+5),B,PSET
430 NEXT
440 SCREEN1,1
450 '****************
460 '** HERE WE GO! **
470 '****************
480 '** TO THE LEFT **
490 '****************
500 XP=230:YP=63
510 GET(1,1)-(8,8),BL,G
520 PUT(XP,YP)-(XP-7,YP-5),B,OR
```

```
530 IFPEEK(&HFF00) AND1 THEN530
540 POKE&HFF23,&H37:SP=JOYSTK(0):POKE&HFF23,&H3F:SP=63-JOYSTK(1)-20
550 AA=3-INT(P/3):IFAA<1 THENAA=1
560 FORI=1TO3:POKE&H7FE5-I,AA*I:NEXT
570 IFS<20 AND SP<0 THEN540
580 IFS<SP THENS=S+1:P=P+S/70:GOTO0600
590 IFS>SP THENS=S-1:P=P-S/70
600 PUT(XP,YP)-(XP-8,YP-8),BL,PSET
610 XP=XP-P
620 GET(XP,YP)-(XP-8,YP-8),BL,G
630 IFS>35 THENGOSUB1040:GOTO650
640 PUT(XP,YP)-(XP-7,YP-5),B,OR
650 IFXP<=20 THENXP=20:GOTO700
660 GOTO540
670 '**********************
680 '** NOW TO THE RIGHT **
690 '**********************
700 YP=127:GET(XP,127)-(XP-7,119),BL,G:GOTO770
710 POKE&HFF23,&H37:SP=JOYSTK(0):POKE&HFF23,&H3F:SP=63-JOYSTK(1)-20
720 AA=3-INT(P/3):IFAA<1 THENAA=1
730 FORI=1TO3:POKE&H7FE5-I,AA*I:NEXT
740 IFS<20 ANDSP<0 THEN710
750 IFS<SP THENS=S+1:P=P+S/70:GOTO0770
760 IFS>SP THENS=S-1:P=P-S/70
770 PUT(XP,YP)-(XP-7,YP-7),BL,PSET
780 XP=XP+P
790 GET(XP,YP)-(XP-7,YP-7),BL,G
800 IFS>35 THENGOSUB1180:GOTO820
810 PUT(XP,YP)-(XP-7,YP-5),B1,OR
820 IFXP>240 THENXP=240:GOTO870
830 GOTO710
840 '**********************
850 '*** THE LAST BIT... **
860 '**********************
870 YP=190:GET(XP,YP)-(XP-7,YP-7),BL,G:GOTO940
880 POKE&HFF23,&H37:SP=JOYSTK(0):POKE&HFF23,&H3F:SP=63-JOYSTK(1)-20
890 AA=3-INT(P/3):IFAA<1 THENAA=1
900 FORI=1TO3:POKE&H7FE5-I,AA*I:NEXT
910 IFS<20 AND SP<0 THEN880
920 IFS<SP THENS=S+1:P=P+S/70:GOTO0940
930 IFS>SP THENS=S-1:P=P-S/70
940 PUT(XP,YP)-(XP-7,YP-7),BL,PSET
950 XP=XP-P
960 IFXP-7<=209 THEN1320
970 GET(XP,YP)-(XP-7,YP-7),BL,G
980 IFS>35 THENGOSUB1040:GOTO1000
990 PUT(XP,YP)-(XP-7,YP-5),B,OR
1000 GOTO880
1010 '*************
1020 '** WHEELIE **
1030 '*************
1040 IFXP<20 THENRETURN
1050 IFS=>40 THEN1070
1060 PUT(XP,YP)-(XP-6,YP-5),B2,OR:RETURN
1070 PUT(XP,YP)-(XP-5,YP-7),B4,OR
1080 FORI=0TO80:NEXT
1090 PUT(XP,YP)-(XP-7,YP-7),BL,PSET
1100 PUT(XP,YP)-(XP-6,YP-4),B9,OR
1110 FORI=0TO125:NEXT
1120 PUT(XP,YP)-(XP-7,YP-7),BL,PSET
1130 PUT(XP,YP)-(XP-7,YP-3),B6,OR
1140 GOTO2060
1150 '***********************
1160 '** WHEELIE OTHER WAY **
1170 '***********************
1180 IFXP>240 THENRETURN
1190 IFS=>40 THEN1210
1200 PUT(XP,YP)-(XP-6,YP-5),B3,OR:RETURN
1210 PUT(XP,YP)-(XP-5,YP-7),B5,OR
1220 FORI=0TO80:NEXT
1230 PUT(XP,YP)-(XP-7,YP-7),BL,PSET
1240 PUT(XP,YP)-(XP-6,YP-4),B9,OR
1250 FORI=0TO125:NEXT
1260 PUT(XP,YP)-(XP-7,YP-7),BL,PSET
1270 PUT(XP,YP)-(XP-7,YP-3),B6,OR
1280 GOTO2060
1290 '****************
1300 '** UP THE RAMP **
```

```
1310 '*****************
1320 XP=209:YP=190:GET(XP,YP)-(XP-7,YP-7),BL,G
1330 FORI=209TO180 STEP-4
1340 PUT(XP,YP)-(XP-7,YP-7),BL,PSET:XP=I:YP=YP-2
1350 GET(XP,YP)-(XP-7,YP-7),BL,G
1360 PUT(XP,YP)-(XP-6,YP-5),B2,OR
1370 FORII=1TO(24-P*2):NEXT
1380 NEXT
1390 X1=XP:Y1=YP
1400 '*****************
1410 '** LIFT OFF... **
1420 '*****************
1430 U=SQR(28000*P/(20+AF))
1440 ST=LOG(YP)*2/TA:X=X+ST:XP=XP-ST
1450 YP=X*TA-(10*X^2/(2*U^2*CA))
1460 YP=180-YP
1470 IFYP=>177 ANDXP>BM THEN1590
1480 PUT(X1,Y1)-(X1-8,Y1-8),BL,PSET
1490 IFXP<1 THEN1500 ELSEIF(PPOINT(XP-1,YP) OR PPOINT(XP,YP)) AND (YP<184 ANDYP>160) THEN1650
1500 IFYP=>184 THEN1900
1510 IFXP<8 THEN1900
1520 GET(XP,YP)-(XP-8,YP-8),BL,G
1530 PUT(XP,YP)-(XP-6,YP-5),B2,OR
1540 X1=XP:Y1=YP
1550 GOTO1440
1560 '*********************
1570 '** BUS TOP LANDING **
1580 '*********************
1590 PUT(X1,Y1)-(X1-8,Y1-8),BL,PSET
1600 PUT(XP,175)-(XP-7,172),B6,OR
1610 GOTO2060
1620 '*************************
1630 '** SUCCESSFUL LANDING! **
1640 '*************************
1650 YP=179
1660 XP=BM-8
1670 GET(XP,YP)-(BM-14,YP-8),BL,G
1680 FORI=XP TO BM-29 STEP-2
1690 PUT(XP,YP)-(XP-8,YP-8),BL,PSET
1700 YP=YP+1:XP=I
1710 GET(XP,YP)-(XP-8,YP-8),BL,G
1720 PUT(XP,YP)-(XP-6,YP-6),B8,OR
1730 FORII=0TO25:NEXTII
1740 NEXTI
1750 PUT(XP,YP)-(XP-8,YP-8),BL,PSET
1760 XP=BM-30:YP=190
1770 GET(XP,YP)-(XP-8,YP-8),BL,G
1780 FORI=0TO10 STEP2
1790 IFXP<10 THEN1860
1800 PUT(XP,YP)-(XP-8,YP-8),BL,PSET
1810 XP=XP-2
1820 GET(XP,YP)-(XP-8,YP-8),BL,G
1830 PUT(XP,YP)-(XP-7,YP-5),B,OR
1840 FORII=0TO40+I:NEXT
1850 NEXT
1860 FORI=1TO350:NEXT:GOTO2300
1870 '****************
1880 '** OVER-SHOOT **
1890 '****************
1900 XP=XP-3:IFXP<8 THENXP=8
1910 YP=186
1920 GET(XP,YP)-(XP-8,YP-8),BL,G
1930 PUT(XP,YP)-(XP-5,YP-7),B4,OR
1940 FORI=0TO220:NEXT
1950 PUT(XP,YP)-(XP-8,YP-8),BL,PSET
1960 XP=XP-3:IFXP<8 THENXP=8
1970 YP=189
1980 GET(XP,YP)-(XP-8,YP-8),BL,G
1990 PUT(XP,YP)-(XP-6,YP-4),B9,OR
2000 FORI=0TO220:NEXT
2010 PUT(XP,YP)-(XP-8,YP-8),BL,PSET
2020 PUT(XP,191)-(XP-7,187),B6,OR
2030 '*************
2040 '** CRASH!! **
2050 '*************
2060 POKE&HFF23,&H37
2070 FORI=0TO350:NEXT
```

```
2080 P=0:X=6:S=0
2090 IFL=0 THEN2170 ELSEL=L-1
2100 CLS0:PRINT@0,CHR$(128);
2110 FORI=1TO4:PCOPYI TOI+4:NEXT
2120 PMODE4,5
2130 GOTO350
2140 '*****************
2150 '** END OF GAME **
2160 '*****************
2170 CLS
2180 IFBN=4 THENBN=0
2190 PRINT:PRINT:PRINT"  YOU HAVE WRECKED ALL OF YOUR    BIKES. YOU MANAGED TO JUMP"
2200 PRINT"  OVER";BN;"BUSES AND LAND."
2210 PRINT:PRINT"    PRESSING 'Q' WILL QUIT.       ANY OTHER KEY WILL GIVE YOU     ANOTHER GO."
2220 I$=INKEY$
2230 IFI$="" THEN2220
2240 IFI$="Q" THENEND
2250 BN=4:L=3:P=0:X=6:S=0
2260 GOTO2100
2270 '*******************
2280 '** GOOD LANDING! **
2290 '*******************
2300 CLS:POKE&HFF23,&H37
2310 PRINT:PRINT:PRINT:PRINT:PRINTBN+1;"BUSES JUMPED SUCCESSFULLY!":PRINT:PRINT:PRINT:PRINT
:PRINT"   PRESS ANY KEY TO CONTINUE"
2320 I$=INKEY$:IFI$="" THEN2320
2330 BN=BN+1:IFBN=12 THEN2380
2340 GOTO2100
2350 '***************
2360 '** FINISHED! **
2370 '***************
2380 CLS
2390 PRINT:PRINT:PRINT:PRINT"  WELL DONE! YOU'VE MANAGED TO     JUMP ALL TWELVE BUSES!"
2400 GOTO2210
2410 '*****************
2420 '** DRAW SCREEN **
2430 '*****************
2440 PCLS
2450 LINE(0,192)-(256,192),PSET
2460 LINE(1,64)-(256,64),PSET
2470 LINE(15,128)-(254,128),PSET
2480 CIRCLE(20,64),20,,1,0.5,0.75
2490 LINE(20,45)-(20,128),PSET:PRESET(20,44):PRESET(0,64)
2500 LINE(20,45)-(25,64),PSET,BF
2510 PAINT(10,50)
2520 LINE(20,128)-(25,109),PSET,BF
2530 LINE(1,64)-(1,109),PSET
2540 CIRCLE(20,109),20,,1,0.25,0.5:PRESET(0,109):PRESET(20,129)
2550 PAINT(10,70)
2560 CIRCLE(235,128),20,,1,0.75,0:PRESET(235,108):PRESET(256,128)
2570 LINE(235,128)-(230,109),PSET,BF
2580 PAINT(250,120)
2590 LINE(235,128)-(235,191),PSET
2600 CIRCLE(235,172),20,,1,0,0.25:PRESET(256,172)
2610 LINE(254,128)-(254,172),PSET
2620 PAINT(240,135)
2630 LINE(43,5)-(214,5),PSET:LINE(43,35)-(214,35),PSET
2640 CIRCLE(43,20),15,,1,0.25,0.75:CIRCLE(214,20),15,,1,0.75,0.25
2650 PAINT(50,20)
2660 LINE(235,191)-(230,172),PSET,BF
2670 LINE(180,191)-(180,176),PSET:LINE-(209,191),PSET:PAINT(190,190)
2680 XX=USR0(3650)
2690 FORI=1TO33
2700 READXV,YV,XW,YW:LINE(XV,YV)-(XW,YW),PRESET,BF
2710 NEXT
2720 FORI=1TO30
2730 READXV,YV,XW,YW:LINE(XV,YV)-(XW,YW),PRESET
2740 NEXT
2750 FORI=5TO8:PCOPY I TO I-4:NEXT
2760 RETURN
2770 '*******************
2780 '** DEF. GRAPHICS **
2790 '*******************
2800 FORY=1TO6
2810 FORX=1TO8
2820 READA:PSET(X,Y,A)
```

```
2830 NEXTX,Y
2840 GET(1,1)-(8,6),B,G
2850 PCLS:XX=USR0(3440)
2860 FORY=1TO6
2870 FORX=8TO1 STEP-1
2880 READA:PSET(X,Y,A)
2890 NEXTX,Y
2900 GET(1,1)-(8,6),B1,G
2910 PCLS
2920 FORY=1TO6
2930 FORX=1TO7
2940 READA:PSET(X,Y,A)
2950 NEXTX,Y
2960 GET(1,1)-(7,6),B2,G
2970 XX=USR0(3450):PCLS
2980 FORY=1TO6
2990 FORX=7TO1 STEP-1
3000 READA:PSET(X,Y,A)
3010 NEXTX,Y
3020 GET(1,1)-(7,6),B3,G
3030 PCLS
3040 FORY=1TO8
3050 FORX=1TO6
3060 READA:PSET(X,Y,A)
3070 NEXTX,Y
3080 GET(1,1)-(6,8),B4,G
3090 XX=USR0(3460):PCLS
3100 FORY=1TO8
3110 FORX=6TO1 STEP-1
3120 READA:PSET(X,Y,A)
3130 NEXTX,Y
3140 GET(1,1)-(6,8),B5,G
3150 PCLS
3160 FORY=1TO4
3170 FORX=1TO8
3180 READA:PSET(X,Y,A)
3190 NEXTX,Y
3200 GET(1,1)-(8,4),B6,G
3210 FORY=1TO15
3220 FORX=1TO10
3230 READA:PSET(X,Y,A)
3240 NEXTX,Y
3250 GET(1,1)-(10,15),B7,G
3260 PCLS
3270 FORY=1TO7
3280 FORX=1TO7
3290 READA:PSET(X,Y,A)
3300 NEXTX,Y
3310 GET(1,1)-(7,7),B8,G
3320 PCLS
3330 FORY=1TO7
3340 FORX=1TO5
3350 READA:PSET(X,Y,A)
3360 NEXTX,Y
3370 GET(1,1)-(5,7),B9,G
3380 PCLS
3390 GET(1,1)-(8,8),BL,G
3400 RETURN
3410 '************************
3420 '** DATA FOR GRAPHICS **
3430 '************************
3440 DATA 0,0,0,0,0,1,0,0,0,0,1,1,1,0,0,0,0,1,0,0,1,0,0,0,1,0,1,1,1,0,1,0,1,0,0,1,0,1,0,
1,0,0,0,0,1,0
3450 DATA 0,0,0,1,1,0,1,0,1,1,0,0,1,0,1,0,1,1,1,0,0,0,1,0,1,1,1,0,0,0,0,0,1,0,1,0,0,0,0,0,1,0
3460 DATA 0,1,0,0,0,0,1,0,1,0,0,0,0,1,0,1,0,0,0,0,1,0,1,0,0,0,1,0,1,0,0,1,1,1,1,1,1,0,1,0,0,
0,0,1,0,0,0,0
3470 DATA 0,0,0,0,1,0,0,0,0,0,0,1,1,1,0,0,0,0,1,0,1,1,0,0,1,1,1,1,1,1,1,1
3480 DATA 1,1,1,1,1,1,1,1,1,1
3490 DATA 1,0,0,0,0,0,0,0,0,1
3500 DATA 1,0,0,0,0,0,0,0,0,1
3510 DATA 1,0,0,0,0,0,0,0,0,1
3520 DATA 1,0,0,0,0,0,0,0,0,1
3530 DATA 1,1,1,1,1,1,1,1,1,1
3540 DATA 1,0,0,0,0,1,0,0,0,1
3550 DATA 1,0,0,0,0,1,0,0,0,1
```

```
3560 DATA 1,0,0,0,0,1,1,1,1,1
3570 DATA 1,1,1,1,1,1,1,1,1,1
3580 DATA 1,1,1,1,1,1,1,1,1,1
3590 DATA 1,1,0,1,0,0,1,0,1,1
3600 DATA 1,1,0,1,0,1,0,1,1,1
3610 DATA 1,1,1,1,1,1,1,1,1,1
3620 DATA 1,1,0,0,0,0,0,0,1,1
3630 DATA 0,1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,1,1,1,0,0,1,1,1,1,0,1,0,1,1,1,0,1,0,1,0,1,0,0,0,0,0,
1,0,0,0,0,0
3640 DATA 0,0,0,0,0,1,0,0,0,0,0,1,0,1,0,1,0,0,1,1,0,1,0,1,1,1,1,0,0,1,1,1,1,1,0
3650 DATA 48,11,58,12,49,20,52,21,44,29,54,30,61,11,75,12,67,13,68,30
3660 DATA 78,11,79,26,91,11,92,26,80,27,81,28,82,28,83,29,84,29,86,30,87,28,88,29,89,27,90,28
3670 DATA 95,11,96,30,108,11,109,30,112,11,126,12,118,13,119,30
3680 DATA 142,11,143,30,144,11,151,12,144,18,151,19,144,29,151,30,159,11,162,12,160,13,161,28,
159,29,162,30,155,23,156,25
3690 DATA 165,11,166,30,167,20,169,21,182,11,183,30,184,11,196,12,184,19,190,20,184,29,196,30
3700 DATA 199,11,200,30,201,11,209,12,201,19,209,20
3710 DATA 47,12,44,15,47,13,44,16,44,15,48,19,44,16,48,20,53,21,58,26,54,21,58,25,57,26,55,28,
57,27,55,29
3720 DATA 79,27,79,27,91,27,91,27,97,12,107,28,97,13,107,29
3730 DATA 152,12,154,14,152,13,154,15,153,15,151,17,153,16,152,17,152,19,155,22,152,20,154,22,
155,26,152,29,154,26,152,28
3740 DATA 169,19,177,11,170,19,178,11,169,22,177,30,170,22,178,30
3750 DATA 210,12,213,15,210,13,213,16,212,16,210,18,212,17,210,19,211,30,202,21,210,30,201,21
```

Interrupt driven sound

When does an interrupt produce a continuous sound? ***Gareth Fenton** explains.*

IN 'Dragon Answers' in a recent issue of *Dragon User*, it was asked if it is possible to produce sound that would play continously while a program runs (as in *Crazy Painter* and the like). The following program is my solution to this problem and works by using the Dragon's IRQ (Interrupt ReQuest) interrupt.

To enter the program, type in the loading program (**Listing one)** and RUN it. If there is an error in the data you will be advised accordingly, and in which line the error lies. Once it is successfully entered, save the data, as a crash caused by any incorrectly typed numbers could result in complete loss of control of your computer, the only course open to you being to switch off and start again.

To use the (saved) program, use the command EXEC &H7500,n. The value of n will determine which of the three functions is to be used.

The options are summarised in the following table:

EXEC &H7500,0 SWITCH TUNE OFF
EXEC &H7500,1 SWITCH TUNE ON
EXEC &H7500,2 DEFINE NEW TUNE

The option 'Define new tune' needs some further explanation. To create the tune to be played use the format:

EXEC &H7500,2,notes (seperated by a comma),0

The number used for each note is the same at that used by the Dragon's SOUND command. For example, as mentioned in the manual, SOUND 89,1 will play Middle C for a short duration. EXEC &H7500,2,89,0 will play Middle C repeatedly until the sound series is changed or until EXEC &H7500,0 is entered.

In order to avoid slowing the 'host' program the note produced is very short. Therefore it may be necessary to repeat the note to make it appear longer.

If you make an error in the definition of a new sound series, a new error message ?FS ERROR (Fenton Sound) will be displayed and any tune currently being played will be switched off. Also, any error in a Basic program which uses this program will automatically switch the tune off as well. The defined series of notes must be less than 255 — the FS error will result if you attempt to define any further notes.

To make a short pause in the tune, the number 255 should be used instead of a note value. The program will accept a variable instead of a number so, by using P=255, the variable P can be used to represent a pause. In the same way, other variables can be used instead of notes, for example C=89 defines variable C as middle C.

A point to remember is that the interrupt driven sound must be switched off (using EXEC &H7500,0) before using the SOUND or PLAY commands.

That is all the information you require to use the program, from Basic. If you wish to use it from a machine code program or are interested in finding out how it works, the complete assembler listing is included and the rest of this article is devoted to its explanation.

As I have already mentioned, the program works by using the Dragon's IRQ interrupt, which occurs every fiftieth of a second. As well as being used to update the basic TIMER, it is also involved in synchronising the video output to the 'frame flyback' of the TV set and so must be used with care.

In order for the program to work, the IRQ vector (the address the IRQ jumps to fifty times a second) must be changed to point to a dispatch routine which calls the new program and then jumps to the ROM routine to handle the interrupt. The vector at $10D is changed to point to @JUMPER, the new dispatch rountine, by the routine @SWITCHON. It is reset to its normal value of $9D3D by the routine @SWITCHOFF.

While it is possible to change the IRQ vector to point to your own routines directly from Basic, it is not advisable as you could well end up losing any program currently stored in memory. It is better to use a machine language routine as I have done — it is much safer, although the IRQ should normally be left alone unless you are sure what you are doing.

Another important part of the program is the routine which actually makes the sound (@SOUND). This makes use of the ROM routine (at address 47776) which sounds a beep for the length in the B register of the pitch stored in location 140 (hex $8C).

In Basic, the IRQ interrupts the playing of notes, resulting in the 'crackling' that is particularly audible during high notes. To produce a cleaner note, it is necessary to turn off or mask the interrupts. This is done by logically ORing the condition code register with 16; they are re-enabled by ANDing the CCR with 239. Unfortunately, this cannot be done from BASIC because the SOUND and PLAY commands use the interrupts to determine how long to play each note. With interrupts disabled, the first note will play indefinitely. To get round this problem, the program has its own loop set up to make the notes independently of the IRQ, which is then disabled. This is done by the subroutine @BEEP in the assembler listing.

The routine @COMMAND determines which option has been selected. It makes use of two ROM routines. The first, CKCOM checks that the next character in the command buffer is a comma carrying on if it is and causing a syntax error if not. The second routine is GETNUM which will return the number or variable in the command buffer as an eight bit value in the B register, causing a function error if the character is not a number or variable name (or is greater than 255).

In order to use this program from your own machine language programs, the routines @SWITCHON and @SWITCHOFF can be called directly. The note values can be stored directly in the sound buffer pointed to by @BUFFADDR, remembering to terminate the string with a zero byte. As is the case when using this program from Basic, a byte value of 255 ($FF) will cause a short pause.

The length of each note played can be altered by storing an alternative value at @PLOOP. This is originally set to 10 and I suggest that if you do change it, you keep the value fairly small as otherwise it will slow down the program too much.

My thanks to Brian Cadge for his very useful Firmware series which detailed some of the ROM routines used in this program.

Listing 1

```
10 'INTERUPT DRIVEN SOUND
20 'LOADING PROGRAM
30 FOR A=&H7500 TO &H7F9F STEP16
40 BLK=BLK+1
50 LC=0
60 FOR L=0 TO 15
70 EXEC&HBBB5
80 READ A$
90 B=VAL("&H"+A$)
100 POKE A+L,B
110 C=C+B
120 LC=LC+B
130 NEXT L
140 READ D
```

```
150 IF LC<>D THEN SOUND1,1:PRINT"DATA ERROR IN BLOCK";BLK:STOP
160 NEXT A
170 IF C<>17020 THEN SOUND1,1:PRINT"THERE ARE AT LEAST TWO ERRORS.
CHECK CHECKS
UMS":STOP
180 PRINT"DATA CORRECT. I SUGGEST YOU SAVETHIS PROGRAM"
190 END
200 DATA 8E,75,23,BF,01,92,86,7E,B7,01,91,1A,10,BD,89,AA,1759
210 DATA BD,8E,51,5D,27,0D,C1,01,27,10,C1,02,27,15,C6,01,1260
220 DATA 7E,83,44,8E,9D,3D,BF,01,0D,39,8D,62,8E,75,4D,BF,1713
230 DATA 01,0D,39,BE,75,9B,34,10,BD,89,AA,BD,8E,51,35,10,1578
240 DATA E7,80,BC,75,9D,2E,D7,5D,26,EC,E7,84,39,8D,03,7E,2139
250 DATA 9D,3D,73,75,98,27,02,8D,01,39,1A,10,BE,75,99,A6,1510
260 DATA 80,27,2B,81,FF,26,07,8D,1C,BF,75,99,20,14,BF,75,1629
270 DATA 99,B7,00,8C,5F,BD,BA,A0,7A,75,97,26,F8,86,0A,B7,2109
280 DATA 75,97,1C,EF,39,10,BE,75,95,31,3F,26,FC,39,BE,75,1830
290 DATA 9B,BF,75,99,39,03,00,0A,FF,00,00,75,9F,76,9E,00,1493
```

Listing 2

```
7500              5      PRT
7500             10 ****************
7500             20 *INTERUPT SOUND*
7500             30 ****************
7500             40 *
89AA             50 @CKCOM EQU 35242
8E51             60 @GETNUM EQU 36433
BAA0             70 @ROMBEEP EQU 47776
8344             80 @ERROR EQU 33604
9D3D             90 @IRQHNDLR EQU 40253
010D            100 @IRQVECT EQU 269
008C            110 @NOTE EQU 140
0191            120 @ERRVECT EQU 401
7500            130 *
7500 8E7523     140 @COMMAND LDX #@SWITCHOFF
7503 BF0192     150      STX @ERRVECT+1
7506 867E       160      LDA #$7E
7508 B70191     170      STA @ERRVECT
750B            180 *
750B 1A10       190      ORCC #16
750D BD89AA     200      JSR @CKCOM
7510 BD8E51     210      JSR @GETNUM
7513 5D         220      TSTB
7514 270D       230      BEQ @SWITCHOFF
7516 C101       240      CMPB #1
7518 2710       250      BEQ @SWITCHON
751A C102       260      CMPB #2
751C 2715       270      BEQ @NEWSTRING
751E C601       280 @FSERROR LDB #1
7520 7E8344     290      JMP @ERROR
7523            300 *
7523 8E9D3D     310 @SWITCHOFF LDX #@IRQHNDLR
7526 BF010D     320      STX @IRQVECT
7529 39         330      RTS
752A            340 *
752A 8D62       350 @SWITCHON BSR @RESTART
752C 8E754D     360      LDX #@JUMPER
752F BF010D     370      STX @IRQVECT
7532 39         380      RTS
7533            390 *
7533 BE759B     400 @NEWSTRING LDX @BUFFADDR
7536 3410       410 @STRLOOP PSHS X
7538 BD89AA     420      JSR @CKCOM
753B BD8E51     430      JSR @GETNUM
753E 3510       440      PULS X
7540 E780       450      STB ,X+
7542 BC759D     460      CMPX @BUFFEND
7545 2ED7       470      BGT @FSERROR
7547 5D         480      TSTB
7548 26EC       490      BNE @STRLOOP
754A E784       500      STB ,X
754C 39         510      RTS
754D            520 *
754D 8D03       530 @JUMPER BSR @START
754F 7E9D3D     540      JMP @IRQHNDLR
7552            550 *
7552 737598     560 @START COM @TOG
7555 2702       570      BEQ @RTS1
7557 8D01       580      BSR @SOUND
7559 39         590 @RTS1 RTS
755A            600 *
755A 1A10       610 @SOUND ORCC #16
755C BE7599     620      LDX @BUFFPTR
755F A680       630      LDA ,X+
7561 272B       640      BEQ @RESTART
7563 81FF       650      CMPA #$FF
7565 2607       660      BNE @CONT
7567 8D1C       670      BSR @DELAY
7569 BF7599     680      STX @BUFFPTR
756C 2014       690      BRA @RTS
756E BF7599     700 @CONT STX @BUFFPTR
7571 B7008C     710      STA @NOTE
7574 5F         720      CLRB
7575 BDBAA0     730 @BEEP JSR @ROMBEEP
7578 7A7597     740      DEC @PLOOP
757B 26F8       750      BNE @BEEP
757D 860A       760      LDA #10
757F B77597     770      STA @PLOOP
7582 1CEF       780 @RTS ANDCC #239
7584 39         790      RTS
7585            800 *
7585 10BE7595   810 @DELAY LDY @COUNT
7589 313F       820 @DLOOP LEAY -1,Y
758B 26FC       830      BNE @DLOOP
758D 39         840      RTS
758E            850 *
758E BE759B     860 @RESTART LDX @BUFFADDR
7591 BF7599     870      STX @BUFFPTR
7594 39         880      RTS
7595            890 *
7595 0300       900 @COUNT FDB $300
7597 0A         910 @PLOOP FCB 10
7598 FF         920 @TOG FCB 255
7599            930 @BUFFPTR RMB 2
759B 759F       940 @BUFFADDR FDB @BUFF
759D 769E       950 @BUFFEND FDB @EBUFF
759F            960 @BUFF RMB 255
769E            970 @EBUFF EQU *
769E            980      END
```

Winners and Losers

Every month Gordon Lee will look at some prize programming

MULTIPLES of 17 were the name of the game in the June competition — the problem being to find the longest possible sequence (or sequences) of multiples such that each number in the sequence must begin with the same three digits in the order that end the preceding number.

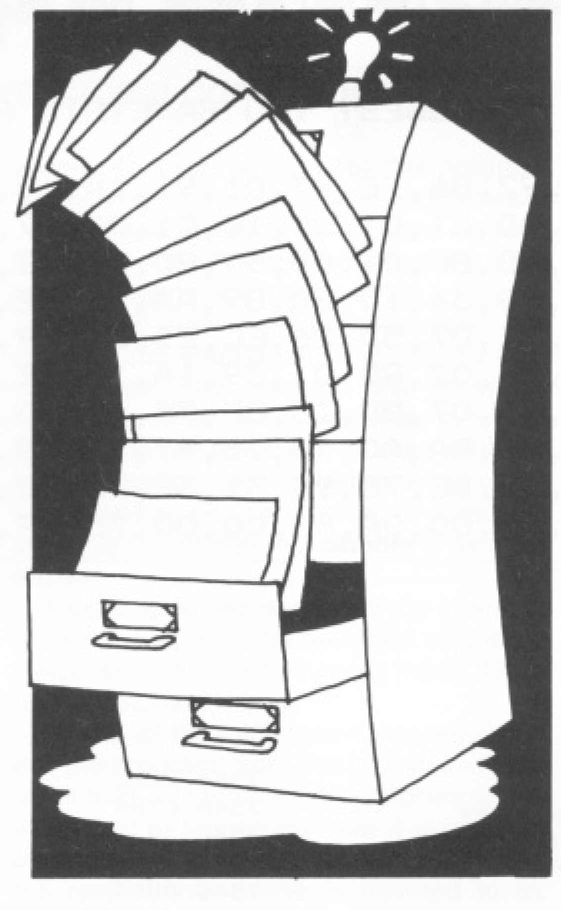
The solution, as given in *Dragon User*, lists two sequences of eight numbers in each. These are shown below, the number in brackets indicating the multiple of 17 at each step.

9095	(535)	9996	(588)
0952	(56)	9962	(586)
9520	(560)	9622	(566)
5202	(306)	6222	(366)
2023	(119)	2227	(131)
0238	(14)	2278	(134)
2380	(140)	2788	(164)
3808	(224)	7888	(464)

Each of these series ends when the final three digits of the last number cannot represent the first three digits of a multiple of seventeen, in the cases above 808 and 888. For example, to take the first of these numbers, the next lowest multiple of 17 that would be in the required range is 9075, with the one above being 8092. Thus, all values in the eight thousand and eighties have been 'jumped'.

These were the results given, but a number of readers came up with a third sequence of eight values commencing with the digits 0901:

0901	(53)
9010	(530)
0102	(6)
1020	(60)
0204	(12)
2040	(120)
0408	(24)
4080	(240)



The acceptance of this as a valid answer depends on the 'leading zero' issue. Philip Beed of Gosport was one of the entrants who picked up this particular point. He writes:

"A simple test for my progress was to use the value you gave to check my results against your own, however, with my first program I got different results even though my program seemed correct. Closer examination revealed that you had included a three digit multiple with a leading zero! So I also included all multiples which could be made four digit by adding leading zeros. This gave me three winning results, only one of which contained no leading zero multiples. I feel there should be only one answer, so pick this one, especially as you say in January that leading zeros are not usually allowed unless specified, and you did not specify it but implied it by including one in your example? I look forward to an answer to the leading zero question."

So the question is, do we allow no leading zero, in which case there is but one solution (9996 *et seq.*), or we allow leading zeros but not in the initial number (giving two solutions, 9095 and 9996), or do we allow unrestricted use of leading zeros to give us three solutions (0901, 9095 and 9996)?

In retrospect, the first option producing a single unique solution would, as mentioned by Mr. Beed, be the most satisfactory — although it was the second option which I had in mind when devising the question. However, fear not. As the question as set contained this ambiguity, all three approaches were considered in selecting the prizewinner. (*Just in case anyone gets the wrong idea, all the prizewinners had produced either the first two solutions, or all three, but no extra points were added for getting the third option, and nobody at all came up with just options two and three, or one and three, because all of you were aware of the leading zero question and had either deliberately included it or deliberately excluded it. Either way, the programming required was of the same calibre — Ed.*).

However, all of this was in danger of being made irrelevant by one solution claiming a sequence of sixteen (yes, sixteen!) numbers. The relevant part of the listing which accompanied this entry is given here, and it provides an interesting excercise in 'reverse' programming.

Can you spot just what the program does and where the programmer went wrong? I'll give you my solution next month.

```
10 REM SEVENTEENS
20 FOR I=1 TO 999
30 L=1:X=I*10
40 R=X-17*INT(X/17)
50 IF R<10 THEN L=L+1:X=10*
   (VAL(RIGHT$(STR$(X),3)))+R:
   GOTO 40 ELSE IF L>M THEN M=L:N=I
60 NEXT I
70 PRINT"The longest sequence starts
   at";STR$(N);", and is";
   STR$(M);" numbers long.":END
```

Dragon Answers

If you've got a technical question write to Brian Cadge. Please do not send a SAE as Brian cannot guarantee to answer individual inquiries.

A basic problem

AS well as my Dragon I also us an Amstrad PC on which I use Locomotive Software's *Basic 2*. While using this I came across two control structures which I would like to add to the Dragon's Basic, as it has several advantages over Basic 2. These are the WHILE . . . WEND and REPEAT . . . UNTIL loops.

Could you please give a simple explanation of how this can be achieved in Dragon Basic?

Adrian Dunn
21 Micklewood Close
Penkridge
Stafford

IT is possible to simulate the control structures you mention using just the IF-THEN and GOTO commands, as follows:

Structured Basic:

```
WHILE X>0 DO
...
Program statements
...
WEND
```

Dragon Basic:

```
100 IF NOT (X>0) THEN 500
110 .....
  Program statements
480.....
490 GOTO 100
500 REM end of loop
```

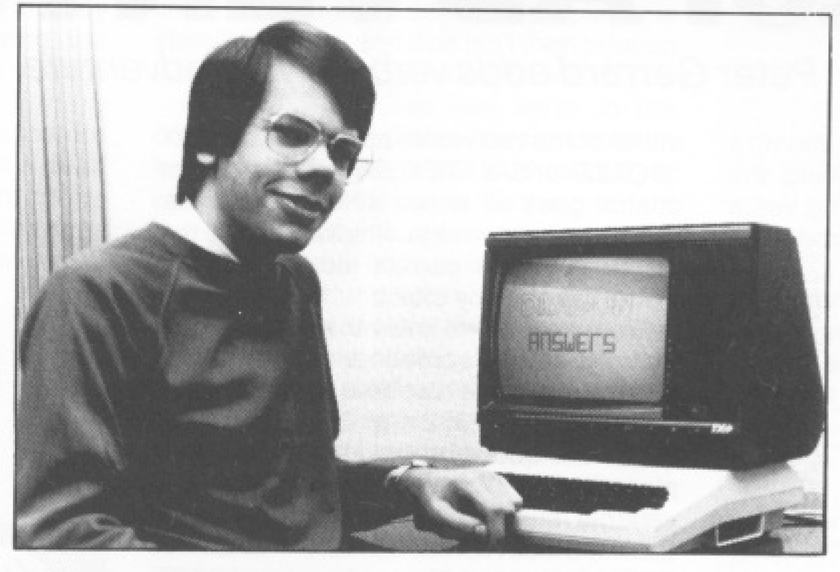
Structured Basic:

```
REPEAT
...
Program statements
...
UNTIL X<0
```

Dragon Basic:

```
100 REM start of loop
110 .....
  Program statements
480 .....
490 IF NOT (X<0) THEN 100
500 REM end of loop
```

Also take a look at BASIC42 from Harris Micro Software with the optional STRUCTUR utility. This will add these control structures directly to the Dragon's vocabulary. Harris can be contacted on 01 570 8335.



Disc drive error

I AM writing to ask your help concerning a problem I have with a disc drive. The interface I am using is the SuperDOS from PNP. The problem is that the write protect error comes up irrespective of the tab being open or closed, so that I cannot format the disc. I have no information on the drive (a Mitsubishi) so I would appreciate your comments please.

David Craig
2 Milebush Close
North Road
Carrickfergus

ASSUMING that the drive you are using is wired correctly to the interface, ie the motor starts and stops, and the busy light works as expected) the problem would seem to be with the write protect line which is pin 28 on the interface.

You can check this by connecting a voltmeter between pins 28 and 1 (or any odd numbered pin which is grounded). Try inserting a write protected disc and a non-write protected disc. The reading should switch between +5V and 0V. if you cannot get such a reading, them the drive is probably at fault.

If all seems well with the drive, then the interface may be at fault, so try contacting PNP directly on 0273 514761 for advice.

Tokenised

PLEASE could you tell me where in the Dragon 32's memory the tokenised Basic program is stored and how to access it. The information I have says it is at $3600 to $7FFF, but when I peek these locations I get a continuation of zeros.

Craig Henderson
39 Woodbury Avenue
Wells
Somerset

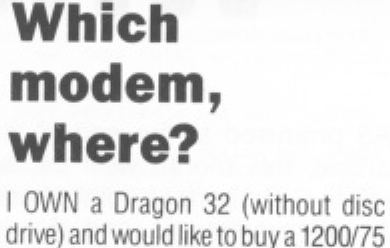
WHERE the tokenised Basic program is actually stored in RAM depends on several things, such as the number of graphics pages reserved and whether DOS is attached or not. You can find the start address of the current Basic program by peeking locations 25 and 26, thus:

```
STADR=PEEK(25)*256+PEEK(26)
```

The end of the program can be found by peeking locations 27 and 28 (which point to the start of the variable table):

```
ENADR=PEEK(27)*256+
  PEEK(28)-1
```

The actual format of the tokenised Basic is quite complicated, and I won't go into it here. Suffice it to say that command and function words are replaced by a single byte in the range >128.

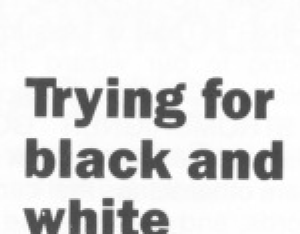


Which modem, where?

I OWN a Dragon 32 (without disc drive) and would like to buy a 1200/75 modem for use with my machine. Could you tell me of a stockist for the above and the cost?

Alastair Scott
26 Marshall Grove
Hamilton
Lanarkshire

IF you intend to use a Dragon 32 with a modem, you will first of all need an RS232 interface. These are available from several suppliers, but do not come cheap, and software may be a problem for the modem you eventually then buy. Your best solution would probably be to pick up a second hand (or discounted) Dragon 64 machine and buy the Hayton Electronics Prism Modem 2000 package (£45). They can be contacted at 36 Laurel Drive, Willaston, South Wirral.



Trying for black and white

I AM trying to get my Dragon to run on a black and white monitor, but unfortunately the Dragon appears only to be set up to run on a colour monitor. Can you tell me how to obtain a signal to make a black and white monitor work?

Chris Findon
8 Welford Road
Shirley
Solihull B90 3HX

THE Dragon's monitor socket supplies composite video (PAL) and sound on pins 3 and 1 respectively (pin 2 is ground). The signal is compatible with monochrome monitors with an impedance of 75 ohms. As you do not say what monitor you are using I can only suggest that you check you have connected the correct pins, and that the monitor is comepatible with the above information.



Write: ADVENTURE

Peter Gerrard adds verbs to your adventure

AS promised at the end of last month's article, this month we'll be tackling the relatively important point of adding verbs to your adventures, and also introducing characters.

Adding verbs is a fairly straightforward task, and since our parser always (if it recognises a verb) returns a value associated with that particular verb, we can then add something like:

```
1000 IF VB=32 THEN 2350
```

or whatever, always knowing that we can of course use the:

```
1000 ON VB GOTO ....
```

command to take up less space.

Assuming that we do that, we then need to introduce the characters. The listing that we're using this month comes from an adventure that I have recently completed writing. One of the characters in it is a wizard of great legend, a chap called Strombrigner the Grey, and he is there to help you in your quest. We'll assume that you've included in your program a line 1000 mentioned earlier, and that the wizard has appeared.

Now then, the syntax which the player is instructed to use in this game, when he's talking to characters that is, takes the form: SAY TO (CHARACTER) "DO SOMETHING". That is, assuming one were talking to our wizard friend and wanted him to GO EAST, one would enter SAY TO STROMBRIGNER "GO EAST", and this is what we have to work on.

There are other things that can be said, magic words, and for these the player is told that he has to use the systax SAY "(MAGIC WORD)". Bearing those things in mind, take a look at the listing.

There are, however, other things that the player can say to the program, and this involves invoking the two magic words that the program will recognise. These are sorted out in lines 2350 and 2351, and if either of the two words are said (as in SAY "POTZZ" or SAY "TETLEY") then program control goes off to two different routines. These just determine whether or not the player is in the correct location for the words to have any effect.

From now on we know that the player is trying to talk to someone, or something, and the rest of the routine is used to sort all this out. Lines 2352 and 2354 determine where the quotation marks appear in the player's input, which then gives us a variable 'talk$'. Thus if the player typed in SAY TO STROMBRIGNER "CAST A SPELL" then 'talk$' would contain "CAST A SPELL", leaving the rest of the string untouched. If, however, the program fails to find any quotation marks (CHR$(£$) by theway) then program execution goes off to line 3864 to print up a suitable response for what this procedure considers to be a nonsensical input.

Having found out what we're saying we then have to find out who we're saying it to, and this is the purpose of the first part of the line 2356, which gives us 'man$' to contain the name of the person being addressed. If, by any chance, the player were to type in SAY TO ME "HELLO" the program would then respond with '"HELLO". There, satisfied?' before going back off to line 10 to get another input. If the player isn't talking to Strombrigner, then line 2357 sends execution careering off to line 2390, which we'll come to later.

Now we know what the player is trying to say and that he's trying to say it to Strombrigner. So, line 2358 checks to see if the wizard is on hand to attempt to carry out the task set before him, and if he isn't then the program kindly informs the player of this fact before going off to line 10 again.

Line 2359 is the start of the main chunk of code for this particular verb, and you'll note that the initial thing that it does is to set the 'ss' variable to 0, indicating that Strombrigner has been spoken to and that he can stop pining for the pub for a while (if you don't give him anything to do then he, reasonably enough, gets bored. Then, if the player has typed in SAY TO STROMBRIGNER "FOLLOW ME" the old wizard peevishly complains that he's here, and we go to line 10 again, which in the particular game is our main control line for the program. All being well, this does keep the wizard by you longer.

Line 2386 is used to tell the player that the syntax is okay so far, that he (the wizard) is thinking about it, and then it sets up a little delay using line 2428 (It's just FOR i=1 TO 3000:NEXT:RETURN). Lines 2361 and 2362 are a miniature parser, and they attempt to unravel the player's request by splitting it up into a verb and a noun. If the player has only entered one word, line 2363 prints up message number 150, which is something to the effect that the



wizard doesn't understands one word requests and you'll have to type in something else. These messages are all printed out via the subroutine at line 5990, which simply accessed a file on disk and prints up on the screen the right message. And, having said that, immediate apologies for using upper and lower case in the listing, but alas my Dragon is not equipped to list out things. Just, as ever, use upper case all the time.

Anyway, there's really only two things the wizard can do, those being to GO somewhere and to CAST a spell, so if either of these are the case we go to line 2387 and 2380 respectively. Otherwise, another little parser comes into play and lines 2365 to 2366 sort out what verb has been entered. If it's a recognised one then we carry on, otherwise print out a message that the wizard is not going to comply and shoot off to line 10. If it is a known verb, and it happens to be verb number 19, 20 or 22 (these being GET, TAKE,and CARRY), then a message about the wizard not being raised by his mother to be a carthorse is printed up. Verb 14, OPEN, gives us a "do it yourself" message, but if none of these conditions are met then the stock message about not complying is used and we retreat, as always, to line 10.

Lines 2380 to 2386 are all about casting a spell, and if the player hasn't typed in CAST A SPELL or CAST SPELL then line 2380 sends him packing with a suitable message. You've got to be in room 37 for the wizard to be able to cast a spell anyway, which is what line 2381 sorts out, and you can't be greedy and have more than one cast for you, which is what line 2382 does.

Lines 2383 to 2384 concern themselves with the number of times that the wizard has visited the pub, and if he hasn't ever been there after your initial meeting then his hands are shaking and he can't do it. If he's been there more than four times then he's too drunk to cope, but all being well he'll cast a spell and open up a new route for you. Thus we increment the score, set the cast spell variable 'cs' and change our

map of the game by altering P%(37,2) to equal 38: this, as you may recall, will allow the player to go south from location 37 to location 38. As location 38 cannot be visited until this is done, we have already set P%(38,1) to equal 37, allowing the player to go out again. Line 2386 is just a theatrical effect!

Now onto the other main reason for Strombrigner's presence, and that is dealt with by lines 2387 to 2389. Without giving away the problem, you basically have to get the old boy to go east at one point, because if you go east you encounter difficulties. Being a wizard, however, Strombrigner can sort things out much better than you can. Bearing that in mind, line 2387 checks to see that the player is specifying east, and if he isn't then print up a response before returning to line 10.



Line 2388 ensures that we're in the correct location and also that the problem hasn't been solved already. If the player isn't in the desired place, or the problem has been dealt with, then we print a message and scuttle back to line 10.

Line 2389, the player's crowning glory, means that we set this particular problem solved variable ('ab'), increment the player's score, print up the relevant message, open up the new route now available to the player, set the solution variable ('bt', as this problem can be approached in two different ways, only one of which works), and we then retreat to line 10 to wait for fresh input.

And that is all that the player can do when talking to characters. This only leaves us with lines 2390 to 2398, used for coping with attempts to communicate with people other than Strombrigner. Line 2390 settles what is contained in 'man$', and then another small sub-chunk of the parser routine we first of all eliminate the unknown (lines 2392 and 2394) before telling the player off for talking to inanimate objects to talk to any character in the adventure other than Strombrigner the Grey. This is, as line 2398 tells you, a straightforward 'You are ignored" message.

The Verb SAY

```
2350 IF MID$(cm$.6,6)="tetley" THEN 6000
2351 IF MID$(cm$,6,5)="potzz" THEN 6010
2352 FOR i=1 TO LEN(cm$):IF MID$(cm$,i,1)=CHR
$(34) THEN talk$=MID$(cm$,i+1,LEN(cm$)-i-1):G
OTO 2356
2354 NEXT:GOTO 3864
2356 man$=MID$(cm$,8,5):man$=LOWER$(man$):IF
LEFT$(man$,2)="me" THEN PRINT x$;talk$;x$:GOT
O 10
2357 IF man$<>"strom" THEN 2390
2358 IF wf=0 THEN PRINT"He's not here.":GOTO
10
2359 ss=0:IF LEFT$(talk$,9)="follow me" THEN
PRINT x$;"I'm here, I'm here!";x$:GOTO 10
2360 PRINT "He's thinking about it ...":GOSUB
 2428
2361 FOR i=1 TO LEN(talk$):IF MID$(talk$,i,1)
=" " THEN vb$=LEFT$(talk$,i-1):na$=MID$(talk$
,i+1):GOTO 2363
2362 NEXT:mess=150:GOSUB 5990:GOTO 10
2363 PRINT:PRINT"He might ...":GOSUB 2428:IF
vb$="go" THEN 2387
2364 IF vb$="cast" THEN 2380
2365 vb$=LEFT$(vb$,5):FOR i=1 TO nv:IF vb$=vb
$(i) THEN vb=i:GOTO 2367
2366 NEXT:mess=7:GOSUB 5990:GOTO 10
2367 IF (vb=19 OR vb=20 OR vb=22) THEN mess=1
11:GOSUB 5990:GOTO 10
2368 IF vb=14 THEN PRINT x$"Do it yourself."x
$:GOTO 10
2369 mess=7:GOSUB 5990:GOTO 10
2380 IF na$<>"spell" AND na$<>"a spell" THEN
mess=10:GOSUB 5990:GOTO 10
2381 IF cp<>37 THEN mess=126:GOSUB 5990:GOTO
10
2382 IF cs=1 THEN mess=127:GOSUB 5990:GOTO 10
2383 IF vp<1 THEN mess=128:GOSUB 5990:GOTO 10
2384 IF vp>4 THEN mess=129:GOSUB 5990:GOTO 10
2385 sc=sc+90:mess=6:GOSUB 5990:cs=1:p%(37,2)
=38
2386 GOSUB 2428:GOSUB 2428:PRINT:PRINT:PRINT
"BOOM!":GOSUB 2428:PRINT:PRINT:PRINT:GOTO 44
2387 IF na$<>"east" THEN mess=130:GOSUB 5990:
GOTO 10
2388 IF cp<>35 OR bt=1 THEN PRINT x$"Not righ
t now."x$:GOTO 10
2389 ab=1:sc=sc+90:mess=110:GOSUB 5990:p%(35,
3)=36:bt=1:GOTO 10
2390 man$=LEFT$(man$,5)
2392 FOR i=1 TO nn:IF man$=na$(i) THEN na=i:G
OTO 2396
2394 NEXT:mess=133:GOSUB 5990:GOTO 10
2396 IF na<>7 AND na<>8 AND na<>9 AND na<>17
AND na<>19 AND na<>30 AND na<>31 AND na<>55 A
ND na<>62 AND na<>34 THEN mess=74:GOSUB 5990:
GOTO 10
2398 PRINT"You are ignored.":GOTO 10
```


Adventure Contact

To help puzzled adventurers further, we are instituting an Adventure Helpline — simply fill in the coupon below, stating the name of the adventure, your problem and your name and address, and send it to Dragon User Adventure Helpine, 12/13 Little Newport Street, London WC2H 7PP. As soon as enough entries have arrived, we will start printing them in the magazine.

Don't worry — you'll still have Adventure Trial to write to as well!

Adventure . . . .

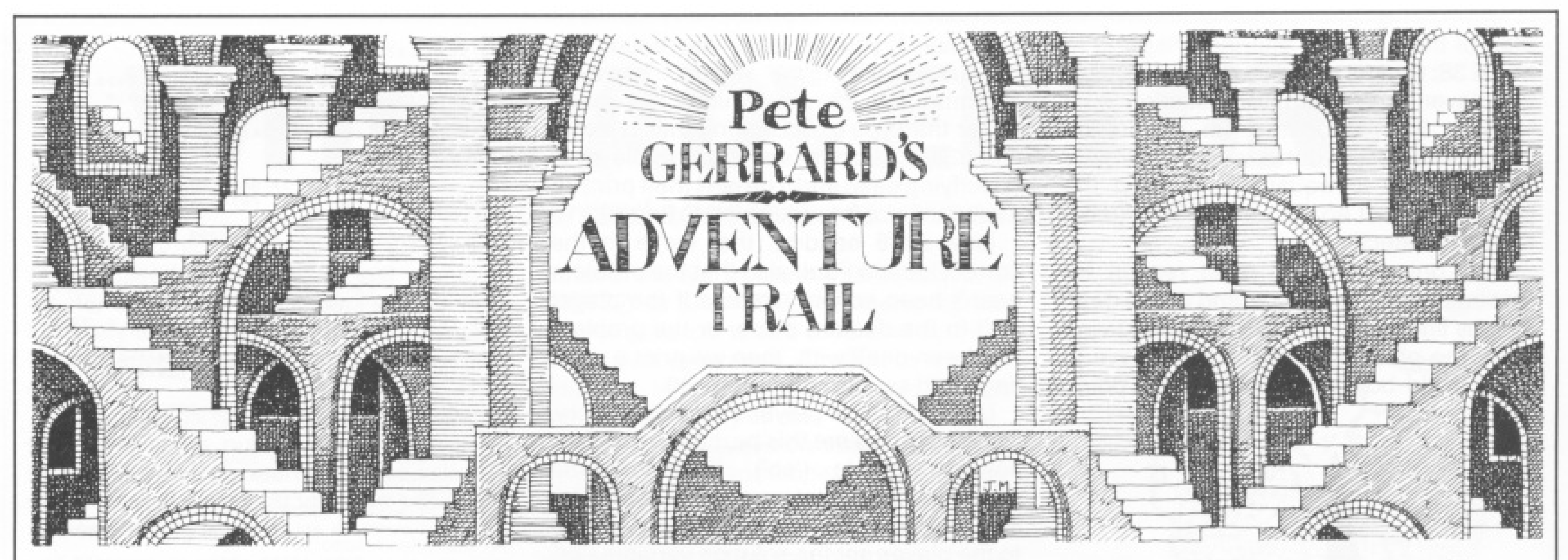
Problem . . . .

. . . .

Name . . . .

Address . . . .

. . . .



AMID malicious lies and rumours put about by scandalmongers and thieves, one's beloved editor forgot to remind yours truly about when the deadline for this month's issue expired. It expires about five minutes after I've finished this article, so to all those people who've sent in tapes this last month (hello Simon and A.H. Beasley and Robert Margrave and someone who's name is indecipherable but could well be Dimli Gloing for all I can make out) many apologies, and over the next week or so I'll be extracting the Dragon from its lair. Then I can dress it all up, dust it all down, and start all over again. All together now . . .

Actually I'm quite looking forward to Robert's game, as he gives a brief description of it in his accompanying letter. Apparently the game is called *The Jewel of Time*, and in it you have to stop time stopping. This, he says, "sounds quite easy but I've included some complications". Sounds easy?! Ye gods, if Einstein had trouble with the stuff (tricky Johnnie, time) then I'm sure that even Trollslayer Pete's going to have the odd problem or two.

Mr. Beesley (he doesn't tell me his first name, must be shy) has a game called *The Temple of Ra*, and to quote "it has taken me a long time to complete it because it started off merely as something to fill in my spare time . . . ". How often have we heard that, fellow adventure writers? I've just completed an adventure of my own (based on that legendary dwarf Dimli Gloing) and I thought I'd set some kind of a record by doing it in a week, writing it initially as a combination of favour for friend and interest in producing a truly bizarre adventure. Four weeks later, and I was still putting in some minor changes prior to writing this. That's the way it goes.

As for Simon Hargrave, the man writes more adventures than Scott Adams. I'll be particularly interested in *The Meaning of Life*, where Simon assures me that the answer is NOT 42, and that you have to control four characters. It would seem that you can make them eat, sleep, wait, drink, and, as he so delicately puts it, "do things to each other". The mind boggles, boggle boggle boggle it goes, and one hesitates to play this game with little ones in the room. We shall see, we shall see.

Why, you may wonder, am I mentioning all these games? Well, ever patient reader, you have noticed that there are precious few adventures being released by commercial companies for your friend the Dragon these days. Indeed, as Simon points out, he hasn't heard of anything since *Tanglewood*, and that is quite some time ago. Consequently any individuals who are releasing games need all the publicity that they can get, and over the next couple of months I'll be doing the best that I can.

Why it should be that companies have deserted the Dragon in such droves is a bit of a mystery to me, since people haven't suddenly all got rid of their computers (apart from a temporary loss for Simon, whose machine blew up) and there is still a large number of Dragons in regular use by adventure fanatics. We'll show 'em!

But enough of this fearless crusade to restore the Dragon as an adventure player's friend, and on to your queries, whinges, whines and moans (a firm of solicitors, perhaps?). First offf the pile this month is a letter from Stephen "I'm really stuck!" Wood, which begins "Dear Pete The Great". that's the way to do it, grovel your way to the top of the heap. Anyway, back to our old friend *Syzygy*. I must have had more letters about this game than any other, but this time a serious complaint. Stephen cannot find the string, and says that it is not in the place that the DU (*Dragon User*, possibly?) map says, or at least not on his version. Well, friend Stephen, we who have the complete solution can reliably inform you that if you follow the following ten steps you'll be well on your way to success, and this should also help a number of other readers who seem to get stuck at step 10. No backwards writing, because it takes too long, and the editor is, not literally one hopes, breathing down my neck.

1) Go to the 3rd computer (not a main one) and PRESS FOUR.
2) Get the Fuzzy Creature.
3) Go to the blank space with the alien creature.
4) Go forward 2 places and THROW FUZZY.
5) Get the light sabre.
6) Go to Vader and defeat him (easy-peasy, I know).
7) Look at the screen of the main computer.
8) Get the blanket and the STRING!!!
9) Go outside the airless corridor.
10) HOLD BREATH and enter corridor.

There, that should sort out a few problems for a collection of you Syzygy-ists. Stephen's next query takes us on to another old favourite, *The Vortex Factor*, which is probably the second most popular (or unpopular, depending on your point of view) adventure for the Dragon. Poor Stephen can't rig up the time machine to work, and so for the benefit of him and anyone else who's stuck at the same point, another collection of steps to take in order to get going, assuming that you've got the yellow and red cartridge and you've got into the cylinder by using the lift. Then . . .

1) Drop the cartridges and the sandwich. Cut the lime and drop the hacksaw. Squeeze the lime, pour the juice then drop the lime peels and the vial.
2) Get the bird and the space suit. Wear the suit and get the coal. Put the coal and the bird into the device. While in the device insert the YELLOW cartridge and press the button.
3) Exit the time machine and examine the bookcase and the counter. Take the pink cartridge and the hairpin. Open the desk and examine it. Read the document to find the combination of the safe. It is 11,29,42 (November 29th 1942).
4) Return to the time machine and drop the hairpin and the document and the pink cartridge. Go to the device. Drop the coal (get it from the time machine) and EXIT. Press the button and return to the time machine.
5) Remove the yellow cartridge and insert the red one.
6) Press the button then EXIT. Open the safe to get a blue cartridge.

Right, that's the story so far, and that is also the time machine up and running and you whizzing about all over the place. Stephen should now no longer be "really stuck!", but let's not neglect the Pete The Great bit, all right?

Where's the next letter? "Dear *Dragon User*" it reads. Oh well, down to earth with a bump again, as usual. An interesting letter this, from someone who will obviously be a doctor one day because their signature is not that good, but I assume it to be F.R. Ellahi. He (or she? Who knows? I'll annoy lots of people and say

that, because the writing isn't neat, F.R. Ellahi must be a male) has been delving around in *Madness and the Minotaur*, and has extracted from it all the messages that are used by the game. I can't print the whole lot, there just isn't room, although anything that includes "by playing the flute according to a song on the parchment a secret passage has been exposed". Obvious, really, something you do every day isn't it, walking about playing flutes and finding secret passages. Anyway, F.R. Ellahi also includes a list of the commands that the program will accept (and points out that the weird ones are spells), so for anyone still trying to plough their way through the game, here they are:

H GET INV DROP BACK JUMP LAMP DRINK EAT FILL ASK LOOK SCORE KILL RUN OPEN HELP CLIMB TIE STAB VETAR MITRA OKKAN AKHIROM NERGAL BELROG CROM ISHTAR PLAY THROW TAKE GRAB QUIET UNCLE

Uncle? Intriguing command. Anyway, thanks for all that lot.

I knew this name would crop up again 'ere long. Paul Iasikiewicz, who will be Paul from now on, asks me to tell all that his address is 40 Sidlaw Adventure, Parr, St Helens, Merseyside WA9 2BQ. And why? Because he has solutions to the following adventures available at 20 pence each, or you can get a little book containing the whole lot for £1.50: *Trekboer, Syzygy, Shenanigans, Lost in Space, Aquanaut 471, Vortex Factor, Black Sanctum, Franklin's Tomb, Calixto Island, Jerusalem Adventure 2, William Adventure 3*. I shall now quote from his letter: "they (you, in other words! P.G.) must enclose an S.A.E. the size or a little bit bigger than *Dragon User*'s Answer Page to the dragon user because they are all big heads who win it." This is it, controversy rears its head once again, all letters of complaint to Paul, not me.

When not slagging people off, Paul has a few hints on *Shenanigans* for you. Once again, I quote . . . "If you can't get the pole into the cave, go into the cave and look for a trap door in the roof. Open the door and go through it. When in the cabin go to the cave and get the pole. Go back to the cabin and then go through the trap door holding the pole". So there you go. And there's more, there's more. In *Syzygy* (again!) he says that "to survive the jump off the cliff just carry the blanket and the string". Finally, in *Aquanaut 471* he says "to get through the force field tell Huey to Go North. When in the cave go E, S, E, E, N, W. Then get the mushy and go up". Mushy! I love it. Mushroom, to you and me. incidentally, at the bottom of Paul's letter is the figure 9. If this is his age, it's astonishing, or perhaps it's just a comment about the IQ of "the big heads who win it". Perhaps we shall find out for next time.

El Diablero is another popular one that was, in its later life, sold without cassette inlay notes, thus causing problems for one or two of you who didn't have the faintest idea of what you were supposed to be doing. Since I haven't got space to print them all now, we'll conclud this month by printing Part One, and provided that I remember (fat chance! I'll try, honestly) we'll start next month with Part Two. The original cassette inlay notes:

"You awake, dazed and confused, in the middle of a desert in the southwest. You had been learning the techniques of sorcery from an old man who lives in these parts. He told you that an evil sorceror, a 'diablero', had become his enemy. Now your teacher is missing and you are alone. Worse still, you can't seem to remember those techniques that you already had learned. The only thing that you can recall is this curious verse:

Remember well the power word,
Remember that which twice you'd heard,
Awake to that which dwells within,
Throw off the yoke of ignorance.

"You have entered the world of El Diablero, where you must learn to acquire and use magical 'power' in order to prepare for the final confronatation with your adversary. *El Diablero* accepts commands in the form of one or two word sentences. For example, LOOK AROUND or GET BOOK are valid commands. Only the first four letters of each word are used. Note: this is by no means an 'easy' adventure. If you are not an experienced adventurer you may want to try a simpler one first . . ."

Me again, that'll have to do for now, part two later, so, from the Wigan Peer, goodbye!

CLASSIFIED ADS

DRAGON 32 for sale. Keys sometimes do not respond properly. Also games, books and 1 Quickshot joystick with auto-fire switch. All for Dragon. Open to offers. Grant Donovan, 17 Chipponds Drive, St. Austell, Cornwall PL25 5DE. Tel: (0726) 75233.

DRAGON 64. With joysticks, books, games and serious software, £125. Also Dragon 32 with joysticks, etc., and back issues of D.U., £75. M. D. Johnson, 25 Radnor Road, Bracknell, Berks. RG12 6QD. Tel: Bracknell (0344) 428699.

DRAGON 32. Graphics tablet; cassette recorder; various books; light pen; Sprite Magic Alldream cartridge; Dreamart 38; Dragon Users; 50 used tapes in racks; all leads inc.; printer cable; joysticks: £250 ono, splits considered. Kevin Hall, 37 Moor Lane, North Hykeham, Lincoln LN6 9AE. Tel:)0522) 692827.

DRAGON hardware and software for sale–cheap! Barry Caruth,' 132 Donaghadee Road, Bangor, Co. Down BT20 4NH. Tel: (0247) 460050.

DRAGON 32 £80. Disc drive: £90. Epson P-40: £30. Joysticks: £5. Complete working system. Many books. All Dragon User. DZSC programs (Orzgzws). R. L. Ruxton, 17 Arran Crescent, Beith, Ayrshire KA15 2DU. Tel: Beith 2898.

DRAGON 32; Sanyo DR101 tape recorder; Sanyo DM2112 green monitor; joystick; many recent games; all leads inc., T.V.; several books; stock control program: £65 ono. Simon McLean, 54 Rayden Crescent, Bolton. Tel: (0942) 813710 eves.

DRAGON 64 with lots of games, utilities, books, joystick and recorder: £70. Disk controller for Dragon: £35. OS-9 operating system with Basic-09 and Pascal, original copies: £40. Or £125 for the lot. N. J. Butters, 11 Underhill Drive, Tonteg, Pontypridd, Mid. Glamorgan. Tel: (0443) 206305.

DRAGON PROGRAMS, books and peripherals for sale from just £1 (swapping may be considered, send your list). For our full list and details, send an S.A.E. to C. Vollbracht, 110 Alpine Rise, Styvechale Grange, Coventry CV3 6NR.

HERE'S MY CLASSIFIED AD.
(please write your copy in capitals on the lines below)

Name . . . .

Address . . . .

. . . . **Tel:** . . . .

Please cut out and send this form to: Classified Department, Dragon User, 12–13 Little Newport St, London WC2H 7PP.

All hands to the wall

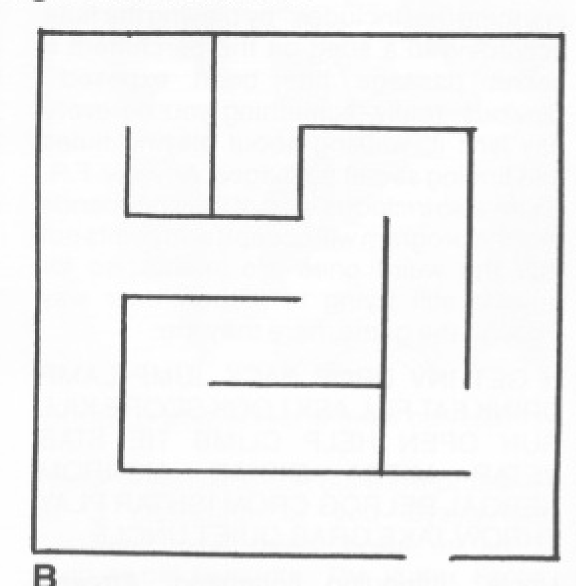
Gordon Lee gets to grips with a series of amazing coincidences

MAZES are probably one of the oldest forms of puzzle. From ancient Greece comes the legend of the labyrinth of King Minos at Crete. This maze-like array of passages housed the Minotaur, a bull-like monster, until it was slain by Theseus, who was then able to escape by following a length of thread which he had used to trace his route.

In the early Christian are maze-like designs were used to decorate the early churches — either in miniature on ecclesiastical vestments, or in the form of a pavement on which a route to the centre could be traced. These mazes were not usually of a puzzle nature, as the route to be taken was generally 'forced' — that is, there was no branching, so the route was simply a convoluted path leading to the centre. Often these were used as a penance to be traversed on hands and knees. A guide book of 1767 describes one such maze at the abbey of St. Stephen at Caen in Normandy:

"...the middle whereof represents a maze or labyrinth about ten feet in diameter, and so artfully contrived that, were we to suppose a man following all the intricate meanders of its volutes, he could not travel less than a mile before he got from one end to the other."

A



B

Mazes in Britain have appeared only within the last 500 years. The earliest of these were known as 'turf mazes' — that is, the paths were simply cut inot the turf. This would allow the solver to be able to trace his route fairly easily, as he could see the overall plan of the maze. The hedge maze became fashionable in the late Renaissance, such as the one at Hampton Court, designed in 1690 for William III. There are not now so many still existing — possibly due to the enormous amount of upkeep which they require — although modern versions, usually with wood or concrete walls, are occasionally encountered. Thse can have an additional feature of gates at certain positions, which can be locked in either the open or closed position, and so allow the plan of the maze to be altered from time to time. The advent of the computer has now added a new dimension to the maze, and there are a number of games of this type available.

Finding the centre of a maze is fairly easy if a plan is available, but otherwise a different strategy must be used. A simple way to avoid being lost in a maze is to keep one hand in contact with the wall as you move about. This must be done from the moment of entering the maze, but it will ensure that you will eventually reach an exit. However, it does not guarantee that you will pass through the centre of the maze. This is dependent on the type of layout — either 'simply' connected, or 'multiply' connected. Maze A is of the simply connected type — that is, all of the dividing walls are directly connected to each other. The 'hand on wall' technique if used on this type of maze will take you twice along every passageway (once in each direction), eventually bringing you back to the entrance. For this reason, most mazes are multiply connected, as in maze B. Here, the technique as outlined *will* bring you back to the entrance, but it is not likely (unless the maze is very badly planned) to take you to the centre.

Next month, we will be looking at the way to solve multiply connected mazes, and in keeping with the tradition of featuring a puzzle game in the Christmas issue, there will be a maze program listing to allow you to put theory into practice!

The competition this month is based on the word 'DRAGON' again. In the inverted

Prize

UP at Preston I met Mr. Rochdale — I mean, up at Rochdale, I met Mr. Preston (they all sound the same to me, these foriggainers), and I said to him, what have you got twenty of this month? And he said, whatever you like. We negotiated BMWs, Cartiers and lobsters, but settled in the end on 20 copies of *Space Trek/Reversi*, the new one from R & AJ Preston, for those of you who can add six up to make 263.

Rules

All in all, it's just another brick in the Dragon. Grab your copy of the OED in 26 volumes, rev up your Dragon and find the words. Then pack up your answer, a copy of your program (printouts only, no cassettes please. We don't mind free cassettes, but the postman's having trouble with his discs), any program notes you wish to add, stuff them in an envelope marked NOVEMBER COMPETITION, and send it off to us here along with your name and address.

I've invented more six-letter words than you've had hot dinners, until my mother-in-law chucked the dictionary at me (that's how I know it's got 26 volumes), so don't try anything with arbity Armstrong, busters, or I'll redefine your definitions for you. What we want for this here tie breaker are some of the six letter words that nobody except you and your Dragon have ever seen before — and we want to know what they mean, too. They *don't* all add up to 263 this time. So if your word looks too familiar, it had better mean something really yabble.

August winners

Somehow winning a prize in the August competition seems just that little bit more — well — *august* than winning in any other month.

The twenty august winners follow, led proudly at the front by none other than Richard Long, who sent not only a program and an answer, but an entire work of art explaining why he wanted a t-shirt. We can't reproduce it, 'cos his colour sense is a bit strange, but here he is anyway:

Richard Long of Camberley, Phil Sapiro of Woolton, Christopher James of Malta, Alan Thomas of Staplehurst, Martin Smith of Lewes, J. Smith of Twyford, Chris Jolly of Orpington (he says it's for his mother in law), Dennis Gates of Durham, J. Pointer of Streatham, F.J. Taylor of Acklam, E.A. Newman of Addlestone, S.A. Siddiqui of Chiswick, A.R. Henderson of Bromsgrove, T. Fawcett of Hendon, Dave Lardener of Rutherglen, M. White of Sheffield, P.D. Maddocks of Maidenhead, C. Hitchinson of Cleveland, G.R. Barber of Sutton Coldfield and S. Wetton of Wimbourne.

Some of the plans for the t-shirts were of a very private nature, but one of the less private ones was that of Phil Sapiro, who is going to save his for the next transvestite party, so that he can go with his drag-on.

Solution

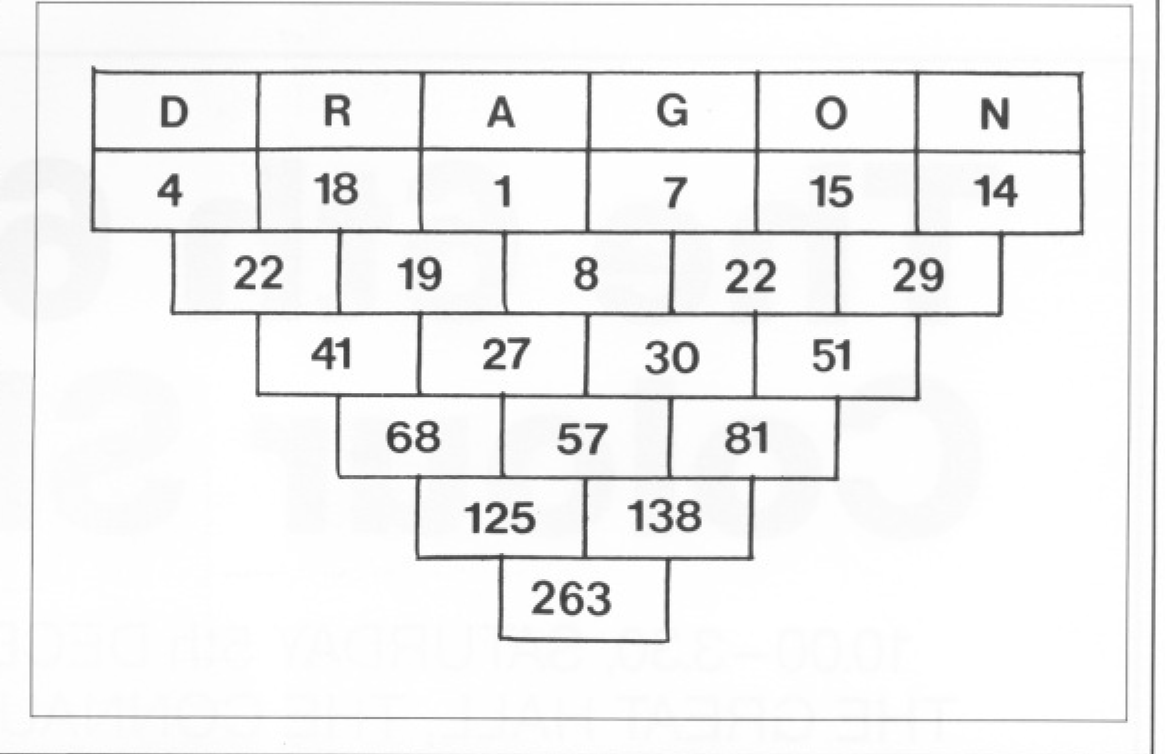
See opposite.

pyramid opposite, each letter has been entered in the top row of 'bricks'. The row beneath contains the alphabetical position of each of the letters in the row above (A=1, D=4, etc.). The brick in each subsequent row is the sum of the two bricks immediately above it, reaching the final total of 263 in the bottom position.

How many other words can you find of six letters in each that also total 263? Preference should be given to common, rather than obscure, words, so the following list, although totalling 263, would not be allowed!

BIJACK — the hijacking of a bicycle
CLEESE — to walk in an unusual manner
COFFLE — a combination of a cough and a snuffle
DRIDEN — a playwright who can't spell

Can you do any better? *(Well, that's about wrapped up the tiebreaker question for this month. See in the box below — Ed.)*



The Answer

This is Gordon Lee's own solution to the August competition
see page 26 for results

ANSWER: The full value of the factor is 188894659314785808547841

SOLUTION: this competition provides a good example of utilising a routine to perform calculations with numbers larger than those which are normally within the arithmetic capability of the computer. The operation is done by storing the actual number being operated on as a *string* variable, and not as a numeric variable. This will allow numbers with over two hundred digits to be stored (even more if some method of splitting the number into sections is devised).

In the progoram, the actual mathematical operations are carried out by the subroutine at line 1000. This routine multiplies a multi-digit number held as a string variable, by a single digit value held as a numeric variable. Before going to the subroutine, the single digit is defined as variable M, and the multi-digit number is entered into string N$. The resulting calculation is to be found in string variable P$ on returning from the routine.

After the initial value of N$ is set, the 75th power of this value is calculated. This is done by multiplying by 2, seventy-five times in the FOR/NEXT loop at line 50, the multiplier M already having been set to two at line 40. Note that at each stage in the loop the value P$, which is returned by the subroutine, is transferred to N$ before the next step in the operation.

The next step is to multiply the resulting total by 5 — once again using the same subroutine. The multiplier M is reset to 5 in line 70, and the subroutine then performs the multiplication.

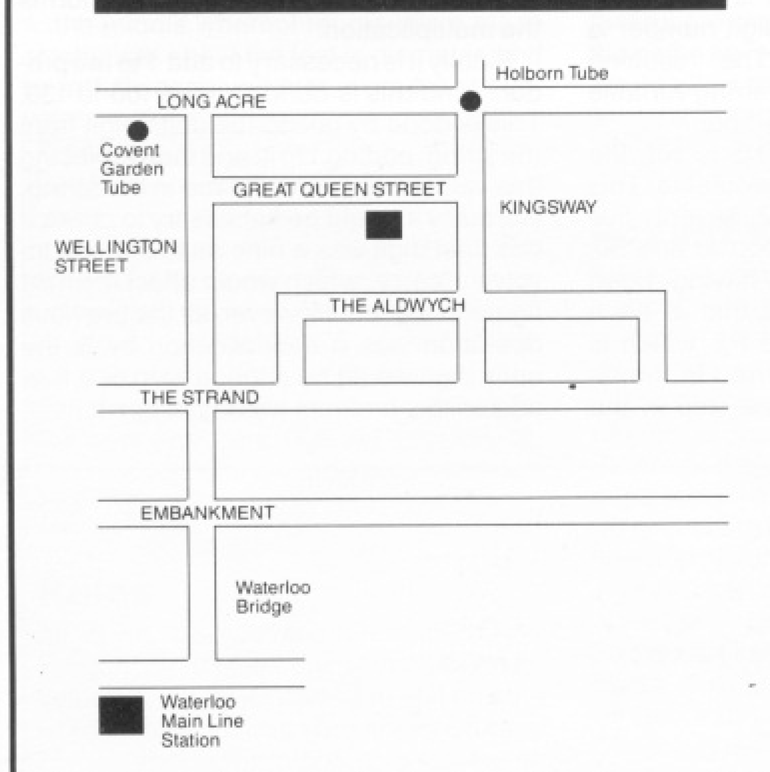
Finally it is necessary to add 1 to the product, and this is done at lines 100 to 130. This is done by taking the units digit from the string, adding 1 to it, and then replacing this value in its correct place in the string. Normally it would be necessary to check if this final digit was a nine as this would involve a 'carry', which would affect the next figure to the left. However, as the previous operation was a multiplication by 5, the units digit would be either a zero or a five, and so this problem would not occur.

```
  10 REM Set up initial value of N$
  20 N$="1"
  30 REM Compute the 75th power of 2
  40 M=2:REM Set multiplier
  50 FOR P=1 TO 75:GOSUB 1000:N$=P$:NEXT
  60 REM Multiply by 5
  70 M=5:REM Reset multiplier
  80 GOSUB 1000
  90 REM Add 1
 100 V=VAL(RIGHT$(P$,1))
 110 V=V+1:V$=STR$(V):V$=MID$(V$,2)
 120 L=LEN(P$)
 130 P$=LEFT$(P$,L-1)+V$
 140 PRINT"The value of 2^75 * 5 + 1 is:":PRINTP$
 150 END
1000 REM Multiplication using Strings
1010 P$="":CARRY=0
1020 FOR D=LEN(N$) TO 1 STEP-1
1030 V=VAL(MID$(N$,D,1))
1040 V=V*M+CARRY
1050 IF V>9 THEN CARRY=INT(V/10):V=V-CARRY*10 ELSE CARRY=0
1060 V$=STR$(V):V$=MID$(V$,2)
1070 P$=V$+P$
1080 NEXT D
1090 IF CARRY<>0 THEN V$=STR$(CARRY):V$=MID$(V$,2):P$=V$+P$
1100 RETURN
```

The 6th 6809 Colour Show

10.00–3.30, SATURDAY 5th DECEMBER 1987
THE GREAT HALL, THE CONNAUGHT ROOMS
GREAT QUEEN STREET, LONDON

new location



The major Dragon and Tandy retailers will be there as well as lots of demonstrations by Dragon users, including the user groups

BUY YOUR TICKETS IN ADVANCE AND SAVE MONEY

Adults **£2.00**
(**£2.50** at door)

Children under 16 **£1.00**
(**£1.50** at door)

To: JOHN PENN DISCOUNT SOFTWARE, DEAN FARM COTTAGE, KINGSLEY, BORDON, HAMPSHIRE GU35 9NG   Tel: Bordon (04203) 5970

Please send me ____________ adults' tickets (at £2.00 each)

and ____________ childrens' tickets (at £1.00 each)

I enclose cheque postal order for ____________

or/ I authorise you to debit my Access account no: ____________   Signed ____________

Name ____________

Address ____________

DR65